Data compression of stereo images and video

This item was submitted to Loughborough University's Institutional Repository by the/an author.

Additional Information:

- A Doctoral Thesis. Submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy of Loughborough University.

Metadata Record: [https://dspace.lboro.ac.uk/2134/10325](https://dspace.lboro.ac.uk/2134/10325)

Publisher: © E.A. Edirisinghe

Please cite the published version.
Please note that fines are charged on ALL overdue items.
Data Compression of Stereo Images & Video

by

Eran Anusha Edirisinghe

A Doctoral Thesis
Submitted in partial fulfilment of the requirements for the award of
Doctor of Philosophy Degree of Loughborough University

11th October 1999

© by E.A.Edirisinghe 1999
Abstract

Data Compression of Stereo Images & Video


One of the amazing properties of human vision is its ability to feel the depth of the scenes being viewed. This is made possible by a process named stereopsis, which is the ability of our brain to fuse together the stereo image pair seen by two eyes. As a stereo image pair is a direct result of the same scene being viewed by a slightly different perspective they open up a new paradigm where spatial redundancy could be exploited for efficient transmission and storage of stereo image data.

This thesis introduces three novel algorithms for stereo image compression. The first algorithm improves compression by exploiting the redundancies present in the so-called disparity field of a stereo image pair. The second algorithm uses a pioneering block coding strategy to simultaneously exploit the inter-frame and intra-frame redundancy of a stereo image pair, eliminating the need of coding the disparity field. The basic idea behind the development of the third algorithm is the efficient exploitation of redundancy in smoothly textured areas that are present in both frames, but are relatively displaced from each other due to binocular parallax. Extra compression gains of up to 20% have been achieved by the use of these techniques.

The thesis also includes research work related to the improvement of the MPEG-4 video coding standard, which is the first audio-visual representation standard that understands a scene as a composition of audio-visual objects. A linear extrapolation based padding technique that makes use of the trend of pixel value variation often present near object boundaries, in padding the exterior pixels of the reference video object has been proposed. Coding gains of up to 7% have been achieved for coding boundary blocks of video objects. Finally a contour analysis based approach has been proposed for MPEG-4 video object extraction.
Acknowledgements

I would like to express my sincere gratitude to my supervisors, Dr. Helmut Bez and Prof. Jianmin Jiang, for their constant encouragement, advice, support, and the patience shown throughout my research programme. I thank my directors of research, Dr. Chris Hinde and Prof. Heiko Schroder (formerly), for their time, effort and suggestions. I would also like to thank fellow research students, Antony, Christos and Antonio for the useful discussions I had with them and for their comradeship.

My thanks are also due to the technical staff members, Mr. Satish Bedi (Loughborough University) and Mr. Kevin Sewell (Glamorgan University), for their kind assistance in technical matters. My sincere thanks also goes to the secretarial staff members, Mrs. Margaret Johnson and Mrs. Judith Poulton of Department of Computer Science, Loughborough University, for their cordiality and assistance with regard to departmental matters, that was always generously forthcoming. My thanks are also due to the academic and secretarial staff of the School of Computing, Glamorgan University.

I would like to express my sincere gratitude to the Department of Computer Science, and Faculty of Applied Science, Loughborough University for the funding provided under my research studentship and for the technical assistance.

A very special thank you goes to the two loves of my life, my wife Gayatri and my baby son Avishka for their love, support, understanding and above all for all the sacrifices made on my behalf. I dedicate this thesis to them with love and gratitude. I am eternally indebted to my parents, parents-in-law and the rest of my family members for their love, encouragement, help and advice that words cannot describe.

Last but not the least, I would like to thank my fellow Sri Lankan, Greek and Indian friends who have made my stay in England and Wales, memorable and enjoyable.

Eran Anusha Edirisinghe

11th October 1999
Contents

Abstract .................................................................................................................. II
Acknowledgements ............................................................................................ III
List of Figures and Tables .................................................................................. IX
Abbreviations & Notations ................................................................................. XII

Chapter 1 - An Overview

1.1 Introduction ................................................................................................ 1
1.2 Motivation for the Research ....................................................................... 3
1.3 General Research Objectives ..................................................................... 4
1.4 Specific Research Objectives ..................................................................... 4
1.5 Research Methodology ............................................................................... 5
1.6 Contributions of Research ......................................................................... 7
  1.6.1 A Pioneering Block Based Predictive Coding Algorithm
       for Stereo Image Compression............................................................... 7
  1.6.2 One Bit Coder for Disparity Field Coding ............................................. 8
  1.6.3 An Object Based Algorithm for Stereo Image Compression ............. 8
  1.6.4 A Contour Analysis Based Technique
       to Extract Objects for MPEG-4 ........................................................... 9
  1.6.5 A Modified Padding Technique
       for MPEG-4 Video Object Planes ..................................................... 10
1.7 An Overview to the Thesis ......................................................................... 10
Chapter 2 - Fundamental Concepts and Literature Review

2.1 Monocular Image Compression ................................................................. 11
2.2 Monocular Image-Sequence / Video Compression ..................................... 12
2.3 JPEG Lossy Compression Algorithm .......................................................... 14
  2.3.1 Discrete Cosine Transform (DCT). ...................................................... 14
  2.3.2 DCT Coefficient Quantization. ............................................................ 17
  2.3.3 Entropy Coding of Quantized DCT Coefficients. .............................. 19
2.4 Stereo Image Compression .......................................................................... 22
  2.4.1 Characteristics of Stereoscopic Signals. ............................................ 22
  2.4.2 Characteristics of Stereoscopic Perception. ........................................ 23
    2.4.2.1 The Suppression Theory. ............................................................ 24
    2.4.2.2 The Fusion Theory .................................................................... 24
  2.4.3 Theory of Stereo Image Coding .......................................................... 24
  2.4.4 Stereoscopic Image Capture ............................................................... 27
    2.4.4.1 Parallel Axis Camera Geometry ............................................... 27
  2.4.5 Stereoscopic Image Display Technology ............................................. 29
  2.4.6 A Comparison of Stereo Pairs and Video Sequences ......................... 30
  2.4.7 Review of Stereoscopic Image Compression Methods ...................... 31
  2.4.8 Stereoscopic Image Sequence Compression ...................................... 37
2.5 Disparity Compensated Transform Domain Predictive Coding Algorithm ....... 43
2.6 Important Definitions ................................................................................ 45
  2.6.1 Extra Compression Ratio (ECR). ........................................................ 45
  2.6.2 The Mean Squared Error (MSE) ........................................................ 45
  2.6.3 The Sum of Absolute Differences (SAD) ............................................ 46
  2.6.4 Peak Signal to Noise Ratio (PSNR) ................................................. 46
Chapter 3 - Block Based Coding of Stereo Image Pairs

3.1 The Use of a First Order Linear Predictor to Improve Prediction .................. 47
   3.1.1 Introduction ....................................................................................... 47
   3.1.2 A and B, Parameter Calculation ........................................................ .49
   3.1.3 Experimental Results and Analysis .................................................. .49
   3.1.4 Conclusions ........................................................................................ 52

3.2 A Novel Coder for Disparity Field Coding .............................................. 53
   3.2.1 Introduction ........................................................................................ 53
   3.2.2 A Mathematical Approach .................................................................. 53
      3.2.2.1 Conventional scheme ............................................................ 54
      3.2.2.2 Proposed scheme .................................................................... 54
   3.2.3 Experimental Results & Analysis ...................................................... 55
   3.2.4 Conclusions ....................................................................................... 59
   3.2.5 Improvements, Applications and Further Research ........................... 59

3.3 Pioneering Block-Based Predictive Coding .............................................. 61
   3.3.1 Introduction ....................................................................................... 61
   3.3.2 Algorithm Design .............................................................................. 62
      3.3.2.1 Algorithm 1 – Single Pioneering Block Coder ............................ 65
         (a) Selection of search window ........................................................ 65
         (b) Selection of pioneering block ....................................................... 66
         (c) Disparity compensation .............................................................. 66
         (d) Prediction .................................................................................... 66
         (e) A mathematical representation .................................................. 67
      3.3.2.2 Algorithm 2 – Two Pioneering Block Coder ............................... 70
   3.3.3 Experimental Results ......................................................................... 71
   3.3.4 Improved Encoder ............................................................................. 75
   3.3.5 Conclusion ......................................................................................... 79
   3.3.6 Improvements, Applications and Further Research ........................... 79
Chapter 4 - An Object Based Algorithm for Stereo Image Pair Compression

4.1 Introduction ................................................................................................... 84
4.2 An Overview.................................................................................................. 85
4.3 Contour Analysis of Stereo Images ........................................................... 88
  4.3.1 Contour Extraction .............................................................................. 88
  4.3.2 Contour Matching ............................................................................. 91
    4.3.2.1 Closed contour matching......................................................... 92
    4.3.2.2 Open contour matching ......................................................... 96
  4.3.3 Contour Blocking ............................................................................. 97
  4.3.4 Block Classification ............................................................... 98
  4.3.5 Contour Filling Algorithm .................................................... 100
4.4 Object Encoding .......................................................................................... 102
  4.4.1 Padding of Left OBR ....................................................................... 103
  4.4.2 Disparity-Based Prediction .................................................... 104
    4.4.2.1 Encoding arbitrary shaped boundary macro blocks............ 104
    4.4.2.2 Encoding Interior macro blocks ......................................... 106
    4.4.2.3 Background Encoding ....................................................... 106
4.5 Experimental Results and Analysis ............................................................ 107
4.6 Conclusions................................................................................................. 110
4.7 Improvements, Applications and Further Research .................................. 110

Chapter 5 - Arbitrary Shaped Video Object Coding in Mpeg-4

5.1 Introduction to MPEG-4 Standard............................................................... 113
  5.1.1 MPEG-4 Video Objects......................................................................... 114
  5.1.2 MPEG-4 Video Object Planes ....................................................... 115
  5.1.3 Motion Estimation and Compensation ......................................... 116
  5.1.4 Texture Coding ............................................................................... 117
  5.1.5 Reference VOP Padding .............................................................. 118
5.2 A Contour Analysis Based Object-Extraction Technique for MPEG-4........ 121
5.2.1 A Survey of Segmentation Based Object Extraction Techniques .... 121
5.2.2 Proposed Technique ........................................................................... 123
5.2.3 Experimental Results ........................................................................ 124
5.2.4 Conclusion & Further Research ....................................................... 128

5.3 Improvements to the VOP Padding Technique of MPEG-4.................. 129
5.3.1 Introduction – The Geometry of Prediction ....................................... 129
5.3.2 Some Important Observations ............................................................ 131
5.3.3 Proposed Solution – Linear Extrapolated Padding ......................... 132
5.3.4 Experimental Results & Analysis ...................................................... 135
5.3.5 Extrapolated Average Padding Technique ....................................... 137
5.3.6 Experimental Results & Observations .............................................. 139
5.3.7 Conclusions & Further Research ....................................................... 140

Chapter 6 - Conclusion ................................................................................. 142

References ........................................................................................................ 145
Appendix I ......................................................................................................... 159
Appendix II - Publications ............................................................................ 160
List of Figures & Tables

Figures

Figure 1.1 Effects of Binocular Parallax ................................................................. 02
Figure 2.1 JPEG Encoder ................................................................................... 14
Figure 2.2 Zigzag Order of Spectral Components .............................................. 16
Figure 2.3 Subjecting an Image Block to JPEG / Inverse JPEG Processes ...... 18
Figure 2.4 JPEG Decoder ................................................................................... 22
Figure 2.5 Parallel Axis Camera Geometry ........................................................ 28
Figure 2.6 Block Diagram of the DCTDP Encoder ........................................... 43
Figure 2.7 Disparity Compensated Predictive Coding ........................................ 44
Figure 3.1.1 Stereo Codec with First Order Linear Prediction ......................... 48
Figure 3.1.2 Results for Stereo Image Pair ‘Lab’ ................................................ 51
Figure 3.2.1 Flowchart of the Novel Disparity Field Coder ................................. 54
Figure 3.2.2 Performance Curves .......................................................................... 57
Figure 3.2.3 Disparity Maps for Stereo Image Pair ‘Cans’ ................................. 58
Figure 3.3.1 Pioneering Block Coder - Encoder .............................................. 63
Figure 3.3.2 Pioneering Block Coder - Decoder .............................................. 63
Figure 3.3.3 The Block Co-ordinate System ......................................................... 65
Figure 3.3.4 Search Procedure - Algorithm 1 ....................................................... 68
Figure 3.3.5 Search Procedure - Algorithm 2 ...................................................... 70
Figure 3.3.6 Experimental Results - Image Pair, ‘Castle’ .................................... 74
Figure 3.3.7 Modified Encoder ............................................................................. 76
Figure 3.3.8 The Decoder ...................................................................................... 76
Figure 4.1  Block Diagram of Object Based Stereo Image Encoder ...............85
Figure 4.2  Left and Right Contour Plots of Image Pair 'Cans' .......................90
Figure 4.3  A Matching Closed Contour Pair .............................................91
Figure 4.4  A Matching Open Contour Pair ...............................................91
Figure 4.5  Chain Code Representation ....................................................93
Figure 4.6  Contour Blocking Procedure ..................................................98
Figure 4.7  Block Classification ..............................................................99
Figure 4.8  Priority of Boundary Macro-Blocks Surrounding an Exterior Block .................................................................103
Figure 4.9  Reconstructed Image Samples ...............................................108
Figure 5.1.1  An Example of a MPEG-4 Video Object ..............................116
Figure 5.1.2  MPEG-4 Padding Process ...................................................118
Figure 5.1.3  Priority of Boundary Macro-Blocks ......................................120
Figure 5.2.1  Block Diagram of the Object Extractor ................................123
Figure 5.2.2  (a) Detected contour & VOP grid (b) Classified blocks: interior & boundary (c) Shape of object .....................................................125
Figure 5.2.3  Two Different Object Extractions .........................................126
Figure 5.2.4  Same VO Chosen from Different Frames ............................127
Figure 5.3.1  Prediction Geometry when the Block to be Encoded is Inside the Reference Frame Block .................................................................129
Figure 5.3.2  Prediction Geometry when the Reference Frame Block Lies Inside the Block to be Encoded .................................................................130
Figure 5.3.3  Interior Pixels & Projected Pixels ...........................................132
Figure 5.3.4  Modified MPEG-4 Padding Process .....................................134
Figure 5.3.5  (a) VOP to be encoded, t = 001 (b) Padded VOP of t = 000, using mpeg scheme (c) Padded VOP of t = 000, using proposed scheme ..........................136
Figure 5.3.6  Error Block Formation in Distorted Boundary Blocks ..................... 138
Figure 5.3.7  Extrapolated Average Padding Technique ................................. 138
Figure 5.3.8  (a) Padded VOP of t = 000 with MPEG-4 technique  
               (b) Padded VOP of t = 000 with Extrap. Ave. technique ................. 140

Tables

Table 2.1  Discrete Cosine Transform Matrix, C ........................................... 16
Table 3.1.1 Experimental Results – Linear Prediction ..................................... 50
Table 3.1.2 Matrix $A$ ................................................................................... 50
Table 3.2.1 Experimental Results – Novel Disparity Field Coder .................... 56
Table 3.2.2 Overhead Bit Compression Ratios ................................................ 56
Table 3.3.1 Compression Performance of Algorithm 2 ................................... 72
Table 3.3.2 Comparing Performance of Algorithm 2 against Algorithm 1 ....... 72
Table 4.1  LoG Mask ...................................................................................... 89
Table 4.2  Experimental Results – Object Based Coder .................................. 107
Table 4.3  Percentage Improvements – Object Based Coder ........................... 109
Table 5.3.1 Experimental Results – Linear Extrapolated Padding .................. 135
## Abbreviations & Notations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D</td>
<td>Two Dimensional</td>
</tr>
<tr>
<td>3D</td>
<td>Three Dimensional</td>
</tr>
<tr>
<td>3DTV</td>
<td>Three Dimensional Tele Vision</td>
</tr>
<tr>
<td>ADPCM</td>
<td>Adaptive Differential Pulse Code Modulation</td>
</tr>
<tr>
<td>CCD</td>
<td>Charge Coupled Device</td>
</tr>
<tr>
<td>CONCOD</td>
<td>Conditional Coder</td>
</tr>
<tr>
<td>CR</td>
<td>Compression Ratio</td>
</tr>
<tr>
<td>DCRC</td>
<td>Disparity Compensated Residual Coding</td>
</tr>
<tr>
<td>DCT</td>
<td>Discrete Cosine Transform</td>
</tr>
<tr>
<td>DCTDP</td>
<td>Disparity Compensated Transform Domain Prediction</td>
</tr>
<tr>
<td>DPCM</td>
<td>Differential Pulse Code Modulation</td>
</tr>
<tr>
<td>EAPad</td>
<td>Extrapolated Average Padding</td>
</tr>
<tr>
<td>ECR</td>
<td>Extra Compression Ratio</td>
</tr>
<tr>
<td>EOB</td>
<td>End of Block</td>
</tr>
<tr>
<td>EPad</td>
<td>Extended Padding</td>
</tr>
<tr>
<td>HMD</td>
<td>Head Mounted Displays</td>
</tr>
<tr>
<td>HRPad</td>
<td>Horizontal Padding</td>
</tr>
<tr>
<td>IQDCT</td>
<td>Inverse Quantized DCT</td>
</tr>
<tr>
<td>JPEG</td>
<td>Joint Photographic Experts Group</td>
</tr>
<tr>
<td>LEPad</td>
<td>Linear Extrapolated Padding</td>
</tr>
<tr>
<td>MPEG</td>
<td>Moving Picture Experts Group</td>
</tr>
<tr>
<td>MRF</td>
<td>Markov Random Field</td>
</tr>
<tr>
<td>MSD</td>
<td>Mean Squared Distance</td>
</tr>
<tr>
<td>MSE</td>
<td>Mean Squared Error</td>
</tr>
<tr>
<td>OBCR</td>
<td>Overhead Bit Compression Ratio</td>
</tr>
<tr>
<td>OBR</td>
<td>Object Bounding Rectangle</td>
</tr>
<tr>
<td>OHD</td>
<td>Off Head Displays</td>
</tr>
<tr>
<td>PSNR</td>
<td>Peak Signal to Noise Ratio</td>
</tr>
<tr>
<td>QDCT</td>
<td>Quantized DCT</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>SAD</td>
<td>Sum of Absolute Differences</td>
</tr>
<tr>
<td>SADCT</td>
<td>Shape Adaptive DCT</td>
</tr>
<tr>
<td>SPT</td>
<td>Subspace Projection Technique</td>
</tr>
<tr>
<td>SQ</td>
<td>Scalar Quantization</td>
</tr>
<tr>
<td>UOFF</td>
<td>Unified Temporal-Spatial Optical Flow Field</td>
</tr>
<tr>
<td>VLC</td>
<td>Variable Length Code</td>
</tr>
<tr>
<td>VLI</td>
<td>Variable Length Integer</td>
</tr>
<tr>
<td>VM</td>
<td>Verification Model</td>
</tr>
<tr>
<td>VO</td>
<td>Video Objects</td>
</tr>
<tr>
<td>VOP</td>
<td>Video Object Planes</td>
</tr>
<tr>
<td>VPIC</td>
<td>Visual Pattern Image Coding</td>
</tr>
<tr>
<td>VR</td>
<td>Virtual Reality</td>
</tr>
<tr>
<td>VRPad</td>
<td>Vertical Padding</td>
</tr>
<tr>
<td>WDC</td>
<td>Windowed Disparity Compensation</td>
</tr>
</tbody>
</table>
Chapter 1 - An Overview

1.1 Introduction

One of the amazing properties of the human vision system is its ability to feel the depth of the scenes being viewed [46]. This judgement of depth enables us to perform day to day activities with precision and ease. Catching a ball thrown at one; crossing a road while judging the speed of vehicles approaching one; picking up that precious mug of coffee in front of one, would not have been reality if our neural network was not created for depth perception and analysis. This is made possible by a process named stereopsis, which is the ability of our brain to fuse together the two images seen by our two eyes (the stereo image pair), to form a single image named, the cyclopean image. It has been found [46] that this image has embedded information about depth and an improved resolution of detail. However, the exact processes that lie behind this complex but, often taken for granted marvel, is not yet known to the medical profession.

It has been known for many years, that under normal viewing conditions, the world appears to us as seen from a virtual eye placed midway between the left and right eye positions. [Hering 1897, Abu Ali Al-Hazan around 1000 AD]. This simple geometrical arrangement has an important consequence: since the world appears different from any of these viewpoints, the image we perceive of the world is never recorded directly by any sensory array, but constructed by our neural hardware! However, it is possible to stimulate our sense of stereo vision artificially by acquiring two pictures of the same scene and then presenting the left image to the left eye and right image to the right eye allowing the brain to fuse them together for three-dimensional perception. This has been the basic principle behind the well-known stereograms that have remained a human entertainment media for some decades.

The slightly different perspectives from which our eyes perceive the world lead to different retinal images, with relative displacements of objects in the two monocular views of a scene. This is known as disparity or binocular parallax. The size and direction of the disparities of an object is a measure of its relative depth. The absolute depth-information can be obtained if the geometry of the imaging system [21,26,50] is known
(see Section 2.4.4.1). The human visual system is able to use these disparities for depth-estimation. Figure 1.1 is an illustration of the effects of binocular parallax on relative displacement of the objects in the two frames.

Figure 1.1, shows that the object closest to the camera image planes (the ball in this case) would have the highest disparity. Note also that the disparities of the three objects are different from each other depending on their apparent distances from the camera image planes [21,26,50].

In the diagram above, it has been assumed that the camera axes are parallel (parallel axis camera geometry, see Section 2.4.4.1) [26]. The principle property of the disparities obtained with such geometry is that the objects or their constituent pixels have no relative vertical displacement. Another condition satisfied by this geometry is that all the objects
in the left frame are displaced relative to the left of the corresponding objects in the right frame. In contrary to this, the camera axis could be arranged such that they converge. [21] (converging axis camera geometry). However, under this geometry the vertical disparities are unlikely to be zero.

1.2 Motivation for the Research

As discussed in the introductory section, a stereo scene can be represented by two constituent images, the left image, which is seen by the left eye (or alternatively in a simulated environment by the left camera) and the right image which is seen by the right eye. If the information pertaining to a stereo scene is to be transmitted or stored, both constituent images would have to be taken into consideration. This is due to the fact that both images are necessary in order to extract the depth information related to the scene. However, this means that in data transmission or storage related to 3D scenes, we would require double the amount of information that would be necessary for a monocular representation of the same scene. As a result, the independent transmission and/or storage of stereo data would not be efficient.

In monocular image compression, the spatial redundancies present in images are exploited efficiently to achieve data compression [141-144,153-157]. In video compression, this exploitation is further extended to the temporal domain [145,146,151,152,169-171,173]. Being monocular images, the constituent left and right images of a stereo image pair could also be compressed on an individual basis in the spatial domain using the intra-frame redundancy present in them. However, as a stereo image pair is a direct result of the same scene being viewed by a slightly different perspective, they open up a new paradigm where spatial redundancy could be exploited. This is the inter-frame redundancy present between the left and right images, which is also named, stereo redundancy in literature.

Although a series of JPEG [141-144] and MPEG [145,146] standards represent a global effort for developing monocular image compression technology, they are not efficient for the compression of stereo image pairs for the simple reason that they have not been designed to exploit stereo redundancy. However, the applications requiring stereo images [2-5,7,24,67,91,98,131] for better information access and interpretation are huge, which
makes stereo image compression technology an important area for further research and development.

The basic philosophy of stereo image compression lies in the solution of the so-called ‘correspondence problem’ [14,27,99]. This is basically the location of points, features, objects or areas in the two frames that correspond to each other. These are then used to exploit the redundancy between the two frames [22,25,29,30,63,64,81-83]. On a broader perspective, the solution to the correspondence problem can be investigated in two ways. They are the intensity-based method [8,15,17,74,118,121-123] and the feature-based method [11,20,21,62,86,96,102-106]. The first looks for correspondence between luminance values. The second determines a set of features in both images and seeks correspondence between them.

Although, several researchers have carried out successful research in the area of stereo image compression [70-73,119-126] and stereo sequence compression [35-38,47,56-60,68,69,78,87-92,93,97,109-112] ample space is still prevalent in the improvement of present algorithms and introduction of novel techniques (see Chapters 2, 3 & 4). This was the main motivation behind this research work.

1.3 General Research Objective

“Design and implement novel algorithms for efficient compression of stereo image pairs”

1.4 Specific Research Objectives for the Thesis

- To design novel approaches to block based stereo image compression.
- To design a novel object based algorithm for stereo image compression.
- To investigate possible improvements and modifications to the disparity map coding of conventional block based stereo image compression algorithms.
- To suggest how the principles of the novel techniques could be used for video coding, i.e. using the techniques in the temporal domain.
• To suggest possible extensions of the proposed stereo image compression techniques to stereo sequence compression, particularly to be used with the existing image and video compression standards such as MPEG-2 [145] and MPEG-4 [146].

• To identify future directions of research.

1.5 Research Methodology

The research work was carried out over a period of 30 months. The areas covered can be broadly categorised into four groups as follows:

(i) Development of a research platform

This involved a literature survey on the existing techniques for stereo image compression (see Sections, 2.4.7,2.4.8) and the identification of a lossy, transform based algorithm to be used as a benchmark [17] in testing the novel algorithms.

The Disparity Compensated Transform Domain Predictive Coding (DCTDP) algorithm [17], proposed by M.G.Perkins in 1992 (see Section 2.5) was found to be the best algorithm that gave optimal results for a wide variety of stereo image pairs, representing, indoor, outdoor and synthetic scenes. This algorithm is widely used in the literature in assessing the performance of stereo image compression algorithms. Thus, it was decided that this algorithm be used as the benchmark. Several different implementations of this algorithm were simulated and tested on a set of standard stereo image pairs obtained from various Internet sites. [148,149].

It was decided that the design and testing of the novel algorithms be limited to stereo image pairs obtained using parallel axis camera geometry for reasons that are explained in detail in Section 2.4.4.1. The experiments were also limited to greyscale image pairs, as colour stereo image pair compression can be approximately regarded as compression of multiple greyscale stereo image pairs. It was also decided to use the left image of the stereo image pair as the reference, solely for the ease of explanation of the theoretical and design aspects. However, the selection of the right frame would be a more practical choice for on line raster scanned stereo image compression [17].
After the above platform was established, the research work was carried out in three stages as follows.

(ii) Developing novel techniques for block-based stereo image compression (Chapter 3)

Block-based techniques have been widely used in the transform domain coding of still images (e.g., JPEG) and video (e.g., MPEG-1, MPEG-2, h263) sequences. The idea was to use these techniques in the compression of stereo image pairs.

The DCTDP algorithm [17] addresses the stereo image compression issue by using the JPEG still image compression standard to block the left image into fixed sized, non-overlapping, sub-blocks, which are later transformed into the frequency domain using a Discrete Cosine Transform (DCT) (see Section 2.3.1). These transformed blocks are then quantized (see Section 2.3.2) to remove the higher frequencies that are insensitive to the human visual system [141], thus obtaining data compression. These quantized blocks are then transmitted after entropy coding (see Section 2.3.3), to achieve further compression. The right image is also blocked into a similar structure as the left. The best possible matches for each of these blocks are found by a windowed search in the left frame. Their displacements (disparities) from the corresponding locations are calculated and are transmitted with the transformed, quantized, prediction error blocks. At the decoder end, as the disparities and errors are available along with the left frame, the right frame could be easily reconstructed.

The basic aim of this research project was to improve the above block-based, DCTDP algorithm to achieve extra compression gains. (see Sections 3.2 & 3.3).

(iii) Developing an object-based stereo image compression algorithm (Chapter 4)

In contrary to blocking the image pair into non-overlapping, fixed sized blocks, and predicting the sub-blocks of the right frame from the left frame, this research intended to divide the images into arbitrary shaped, closely matching objects / areas and to use the associated redundancy in achieving data compression. The algorithm has been developed in a similar spirit to MPEG-4 [146], in which an object based coding strategy is used in
exploiting inter-frame redundancy. This object based approach gives greater performance in terms of compression, flexibility and user interactivity.

(iv) Applications of the developed principles in MPEG-4 video coding (Chapter 5)

The aim of this research project was to use the techniques developed in conjunction with the object-based stereo image compression algorithm, and their modifications, in improving the MPEG-4 multimedia standard.

1.6 Contributions of Research

The following original contributions have been made. The resulting conference and journal papers are included in Appendix II, and are referenced as A1, A2, ... A10.

1.6.1 A Pioneering Block Based Predictive Coding Algorithm for Stereo Image Compression

An effective block-based predictive coding algorithm to achieve substantial extra data compression for encoding right frames of stereo image pairs has been proposed (see Section 3.3). [A1,A5].

The basic philosophy of the algorithm involves constructing a pioneering block in the right frame and conducting a windowed search for its best match in the left frame (see Section 3.3.2.1). The pioneering block is selected as the block to the immediate left of the block to be encoded. A predictive block is then found by exploiting both inter-frame and intra-frame correlation to produce predictive errors for further compression. The predictor is the block on the immediate right of the best match of the pioneering block, in the left frame. Due to the special pioneering block based disparity compensation technique used, the disparity vectors need not be transmitted as overhead bits. This results in extra data compression.

A second algorithm based on the same principle has been proposed and analysed. It uses two pioneering blocks to search the left frame window (see Section 3.3.2.2). The two blocks involve the one to the immediate left and the one immediately above the block to
be encoded. Results indicate that this algorithm performs better than the first in terms of compression efficiency and reconstructed right image quality.

Extensive experiments indicated that the above algorithms were sensitive to JPEG compression quality. Further refinements to the above algorithms were done by introducing a JPEG decoder, at the encoder end (see Section 3.3.4) [A2,A4]. This enabled the search for the best match to be conducted on the locally reconstructed left frame, making the predictive coding identical at the encoder and decoder ends. A feedback loop was added to the encoder end so that the selected pioneering blocks would also be in the reconstructed domain at the time of disparity compensation.

1.6.2 One Bit Coder for Disparity Field Coding

A novel method of coding the disparity values of matching sub-blocks in block based stereo image compression algorithms have been proposed and analysed (see Section 3.2). [A10].

The proposed coder makes use of the high probability of two adjacent blocks having equal disparity values, to achieve an overall reduction in the overhead bits. If the block directly above a block to be encoded has the same disparity value, only one bit is used to indicate that the present disparity is the same as in the previous block. If not, an additional amount of bits are used to indicate the new disparity value. A mathematical proof under which this technique would be found advantageous has been formulated.

1.6.3 An Object Based Algorithm for Stereo Image Compression

A novel object-based algorithm for the compression of stereo image pairs has been proposed (see Chapter 4). [A7,A9].

The algorithm effectively combines the simplicity and adaptability of the existing block based stereo image compression techniques with an edge/contour based object extraction technique to determine appropriate compression strategy for various areas of the right image. Extensive experiments carried out support that significant improvements of up to 20% in compression ratio can be achieved by the proposed algorithm, compared with the existing stereo image compression techniques. Yet the reconstructed image quality is
maintained at an equivalent level in terms of peak signal to noise ratio (PSNR) values. In terms of visual quality, the right image reconstructed by the proposed algorithm does not incur any noticeable effect compared with the outputs of the best algorithms.

The proposed algorithm performs object extraction and matching between the reconstructed left frame and the original right frame to identify those objects that match but are displaced by varying amounts due to binocular parallax. Different coding strategies are then applied separately to the internal areas and the bounding areas for each identified object. Based on the mean squared matching error of the internal blocks and a selected threshold, a decision is made whether or not to encode the predictive errors inside these objects. The boundary blocks are matched using a novel shape adaptive, boundary block padding technique to achieve data compression. This method optimises the matching and minimises the amount of bits necessary to encode the boundary error blocks. Its output bit stream includes entropy coding of object disparity, block disparity and possibly some errors, which fail to meet the threshold requirement in the proposed algorithm.

In addition to the above the following contributions, which are outside the area of stereo image compression, have also been made. They are a direct result of the extensions, modifications and observations made with regard to the above object based stereo image compression algorithm.

1.6.4 A Contour Analysis Based Technique to Extract Objects for MPEG-4

A contour analysis based technique has been proposed as an alternative to segmentation based techniques, for MPEG-4 video object extraction (see Section 5.2). [A6]

Firstly, the object contours are extracted by convolving a given frame with a Laplacian-of-Gaussian operator, followed by an edge detection process. The contours are later blocked and classified into three categories in compliance with MPEG-4 standard. A pixel-based parity check, contour filling algorithm is used in identifying the pixels that are within and outside the contour.
1.6.5 A Modified Padding Technique for MPEG-4 Video Object Planes

A linear extrapolation based padding technique has been proposed to minimise prediction errors of arbitrarily shaped boundary blocks in MPEG-4 video object planes (see Section 5.3). [A3,A8].

The proposed technique makes use of the trend of pixel value variations present in boundary macro-blocks to reduce prediction errors. Experimental results indicate that coding gains of up to 7% can be achieved in the coding of arbitrarily shaped boundary blocks as a result of this modification. The feasibility of using several other modifications has also been investigated.

1.7 An Overview to the Thesis

The thesis is divided into six chapters.

Chapter 1, introduces the reader to the subject area of stereovision and discusses the importance and need of compressing stereo image pairs. It gives an overall picture of the work carried out and a summary of contributions made.

Chapter 2, introduces the reader to the basic concepts associated with the algorithm designs that are discussed in the subsequent chapters. It also includes a literature survey on work carried out by other researchers in stereo image pair and image sequence compression.

Chapters 3,4 and 5, include information on original contributions made to the field of stereo image compression and related fields by the author. Each of these chapters concludes with a discussion on the contributions made by the proposed algorithms and possible improvements.

Chapter 6 concludes with an insight into the future directions of research in stereo image pair and video sequence coding.
Chapter 2

Fundamental Concepts and Literature Review

This chapter introduces the reader to the principle research topic, starting from the fundamentals of monocular image and video compression. Important concepts covered include: monocular image compression techniques (Section 2.1), monocular image-sequence/video compression techniques (Section 2.2), JPEG (Joint Photographic Experts Group) image compression standard (Section 2.3), how monocular compression techniques can be extended in compressing binocular (stereo) image and video sequences for efficient transmission and storage (Section 2.4), stereo image capture (Section 2.4.4) and display (Section 2.4.5) technology, a survey into existing technologies (Section 2.4.7) and finally an extensive review of the benchmark algorithm [17] (Section 2.5), which is used to assess the performance of novel techniques proposed in this thesis.

2.1 Monocular Image Compression

Image compression methods, a sub-field of general data (text, audio, image & video etc.) compression methods, are straightforwardly categorised into two groups, namely, ‘lossless’ and ‘lossy’.

Lossless image compression methods [153-157,172,176,177] work as images and image streams (e.g. video) have internal order, i.e. similarity relationships between nearby (and sometimes not so nearby) parts. Thus, they have parts that are predictable from other parts. Describing, rather than reproducing the redundancy that this predictability implies, can save storage space and transmission bandwidth. JPEG-LS [144] is a typical example of lossless image compression standard. However only typical compression ratios of around 2:1 to 3:1 can be achieved by these techniques.
Lossy compression methods work on the principle that similar parts can be ‘reasonably well approximated’ by one typical part, turning similarity into redundancy. Then the approximated image can be losslessly (e.g. entropy coding [141,150,172]) compressed. For example, the huge dynamic range of intensities in a photographic negative can, as far as the human visual system is concerned, be reasonably well approximated by 256 grey levels (8 bits/pixel). A tricolour image that starts out as red, green and blue sub-images of 8 bits/pixel each (24 bits/pixel total), can be reasonably well approximated by a palette of 256 symbols (only 8 bits/pixel) each of which stands for a 24 bit colour that exactly or approximately appears prominently in the image. With good lossy compression algorithms, reasonable visual quality imagery can be stored and transmitted using a few tenths of a bit per pixel (80:1). A typical image compression standard that represents this category is Lossy JPEG [141-143].

Summarising the above, it is seen that both lossless and lossy image compression techniques, exploit similarities between adjacent pixels to achieve data compression in transmission and in storage with the difference being in the reconstructed image quality. This spatial domain (within the area of a given image) redundancy exploitation is named as intra-image coding.

2.2 Monocular Image-Sequence / Video Compression

An image sequence is a collection of digitised snapshots of a scene, taken at fixed time intervals in the temporal domain (time scale).

In typical image sequences, background objects do not generally move from one frame to the next, and only a small percentage of the scene undergoes motion. The displacement of these moving objects usually does not exceed a few pixels. Thus, a lot of redundant information exists between adjacent images (frames) of a video sequence. This redundancy is named as inter-frame redundancy. Video compression techniques and standards exploit inter-frame redundancy, along with the intra-frame redundancy present in the spatial domain of individual frames, to achieve data compression.
The most popular broadcast quality digital television and video coding standards, such as MPEG-1, MPEG-2 [145], H261 and H263 take advantage of the fact that corresponding pixels typically change only slightly from frame to frame. Thus, intra-coded (i.e. complete, I frames) frames need be transmitted only occasionally (1-2 per second in MPEG-1/2, much less in H 261/263). The inter-coded (P, predicted & B, bi-directional frames) frames can be synthesised at the receiver from the intra-coded frames, plus a few dynamic coefficients (motion vectors) that tell the receiver how to estimate and interpolate the motion dynamics, plus perhaps also a residual (reconstruction error) image stream. This residual error is coded compactly because it's amplitude distribution is typically much narrower than that of the original image (the entropy of the residual image is much smaller than the entropy of the original image). As a result large compression factors can be achieved losslessly, by assigning the shortest codes, the most probable amplitudes [172]. The insensitivity of the perceived result to occasional large glitches in the coding makes it possible for lossy methods to provide additional compression factor gains.

It was mentioned that the typical compression factors that can be achieved using lossless compression strategies are limited to 2:1 to 3:1. Thus, in applications that require higher compression gains, the use of lossy compression algorithms to exploit redundancy and reduce bandwidth requirements or storage space is inevitable.

The lossy JPEG [141-143] standard is the most widely used algorithm for the compression of still images and has been used as the basic compression algorithm in the work proposed in latter chapters. Section 2.3 describes the algorithm in detail. Particular emphasis has been given to the following concepts associated with this standard [142,143].

*Discrete Cosine Transform (DCT)* (Section 2.3.1) – A lossless mathematical transformation which is the key to the compression process.

*Scalar Quantization (SQ)* of DCT coefficients (Section 2.3.2) - This is the reason for the lossiness of the JPEG compression algorithm.
Entropy Coding (Section 2.3.3) – A lossless compression scheme which further exploits the redundancy on the data that results from the above two processes.

2.3 JPEG Lossy Compression Algorithm [141-143]

This algorithm operates in three successive stages as shown in Figure 2.1. At the input to the encoder, source image samples are grouped into \( N \times N \) blocks (\( N=8 \) for JPEG), shifted from the unsigned integers with range \([0, 255]\) to signed integers with range \([-128, +127]\) and input to the DCT transformation stage.

2.3.1 Discrete Cosine Transform (DCT)

The key to the compression process discussed in this thesis is the lossless mathematical transformation, the Discrete Cosine Transform (DCT) [150]. It converts spatial (pixel domain) information into frequency (spectral) domain information, where the psycho-visual properties of human vision and spatial redundancy can be combined to achieve data compression [142-143]. As the human eye is not sensitive to high frequency luminance variations, these can be removed from the spectral representation to achieve data compression. This is done by dividing the higher frequency components by larger integer values, in the scalar quantization step (see Section 2.3.2), which follows.
The forward and inverse discrete cosine transforms are defined as follows [150]:

\[
\text{Pixel}(x, y) = \frac{1}{\sqrt{2N}} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} C(i)C(j) \text{DCT}(i, j) \cos \left( \frac{(2x+1)i\pi}{2N} \right) \cos \left( \frac{(2y+1)j\pi}{2N} \right) \quad (2.1)
\]

\[
\text{DCT}(i, j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \text{Pixel}(x, y) \cos \left( \frac{(2x+1)i\pi}{2N} \right) \cos \left( \frac{(2y+1)j\pi}{2N} \right) \quad (2.2)
\]

where,

\[
C(x) = \begin{cases} 
\frac{1}{\sqrt{2}} & \text{if } x = 0 \\
1 & \text{if } x \geq 1
\end{cases}
\]

\text{Pixel}(x, y) \text{, is the pixel value at location } (x,y) \text{ on a } N \times N \text{ square matrix of the image and } 0 \leq i, j \leq N-1.

In matrix form, the transformations can be defined as follows [142,143,150]. First the Cosine Transform matrix \( C \) is defined as,

\[
C_{ij} = \begin{cases} 
\frac{1}{\sqrt{N}} & \text{if } i = 0 \\
\frac{2}{N} \cos \left( \frac{(2i+1)j\pi}{2N} \right) & \text{if } i > 0
\end{cases} \quad (2.3)
\]

where \( 0 \leq i, j \leq N-1 \). If \( B \) defines an \( N \times N \) image block, its forward Discrete Cosine Transform, \( \text{DCT} \) is defined in matrix form as,

\[
\text{DCT} = C \ast B \ast C^T \quad (2.4)
\]

where, \( C^T \) is the transpose of the cosine transform matrix \( C \) and \( \ast \) defines a matrix multiplication. The inverse discrete cosine transform is defined as,
\[ IDCT = C^T \ast DCT \ast C \] (2.5)

Using equation 2.3 above, the Cosine Transform Matrix, \( C \), for \( N=8 \), is calculated as,

\[
\begin{array}{cccccccc}
0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 \\
0.4904 & 0.4157 & 0.2778 & 0.0975 & -0.0975 & -0.2778 & -0.4157 & -0.4904 \\
0.4619 & 0.1913 & -0.1913 & -0.4619 & -0.4619 & -0.1913 & 0.1913 & 0.4619 \\
0.4157 & -0.0975 & -0.4904 & -0.2778 & 0.2778 & 0.4904 & 0.0975 & -0.4157 \\
0.3536 & -0.3536 & -0.3536 & 0.3536 & 0.3536 & -0.3536 & -0.3536 & 0.3536 \\
0.2778 & -0.4904 & 0.0975 & 0.4157 & -0.4157 & -0.0975 & 0.4904 & -0.2778 \\
0.1913 & -0.4619 & 0.4619 & -0.1913 & -0.1913 & 0.4619 & -0.4619 & 0.1913 \\
0.0975 & -0.2778 & 0.4157 & -0.4904 & 0.4904 & -0.4157 & 0.2778 & -0.0975 \\
\end{array}
\]

Table 2.1 Discrete Cosine Transform Matrix, \( C \)

The result of converting an \( N \times N \) image block into its frequency domain, is a \( N \times N \) array of frequency components. They are arranged in the array in such a way that they represent the magnitude of spectral components arranged in an ascending order of frequency starting from the top, left hand corner of the array continuing in a zigzag order to the bottom right hand corner. Figure 2.2 shows the order for a \( N \times N \) array. The component at \((0,0)\) is the magnitude of the DC coefficient (i.e. zero frequency component). It represents the average of the pixel values inside the \( N \times N \) array. All other coefficients are AC (i.e. non-zero frequency) coefficients.

![Figure 2.2 Zigzag Order of Spectral Components](image)
2.3.2 DCT Coefficient Quantization

The coefficient values of the DCT transformed $N \times N$ matrix are in the range $-2^8$ to $2^8-1$. These values are scaled down in order to reduce the number of bits necessary to code all the coefficients. However, the scaling factor may not be equal for all the coefficients. This is because the human eye is not sensitive enough for the very high frequency information [142-150]. Thus, the coefficients representing the higher frequencies can be scaled by larger values. Therefore a scalar quantization matrix $Q(i,j)$ is used for the quantization of the transformed matrix. The quantization and de-quantization formulae are as follows.

\[
QDCT(i,j) = \text{round}\left[\frac{DCT(i,j)}{Q(i,j)}\right]
\]
and,

\[
IQDCT(i,j) = Q(i,j) \times QDCT(i,j)
\]

where, $QDCT(i,j)$ is the quantized transformed matrix and $IQDCT(i,j)$ is the inverse quantized transform matrix. The rounding-off (round) is done to the nearest integer and is the reason for lossiness of the JPEG compression algorithm.

The quantized DCT coefficients are subsequently entropy coded before transmission. This is discussed in detail, in the following section [2.3.3].

The effects of the DCT and the scalar quantization stages on a typical image block are best explained using an example.

Figure 2.3 shows an example of subjecting an $8 \times 8$ image sub-block to the forward and inverse DCT processes and scalar quantization / de-quantization procedures discussed in the previous section. Note, that the sub-block would be reconstructed with loss of quality. Figure 2.3(c) shows the luminance quantization matrix [141] used in the lossy JPEG standard. Note the apparently ascending magnitudes of the quantization values are in zigzag order. The human eye's sensitivity to various frequencies have been taken into account in the standardisation of this table.
<table>
<thead>
<tr>
<th>Source Image Sub-block</th>
<th>Forward DCT Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>16 11 10 16 24 40 51 61</td>
<td>15 0 -1 0 0 0 0 0</td>
</tr>
<tr>
<td>12 12 14 19 26 58 60 55</td>
<td>-2 -1 0 0 0 0 0 0</td>
</tr>
<tr>
<td>13 13 16 24 40 57 69 56</td>
<td>-1 -1 0 0 0 0 0 0</td>
</tr>
<tr>
<td>14 17 22 29 51 87 80 62</td>
<td>0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>18 22 37 56 68 109 103 77</td>
<td>0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>24 35 55 64 81 104 113 92</td>
<td>0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>49 64 78 87 103 121 120 101</td>
<td>0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>72 92 95 98 112 100 103 99</td>
<td>0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>Quantization Table (Standard JPEG)</td>
<td></td>
</tr>
<tr>
<td>240 0 -10 0 0 0 0 0</td>
<td>144 146 149 152 154 156 156 156</td>
</tr>
<tr>
<td>-24 -12 0 0 0 0 0 0</td>
<td>145 148 150 152 154 156 156 156</td>
</tr>
<tr>
<td>-14 -13 0 0 0 0 0 0</td>
<td>155 156 157 158 158 157 156 155</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0</td>
<td>160 161 161 162 162 159 157 155</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0</td>
<td>163 163 164 163 162 160 158 156</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0</td>
<td>163 164 164 164 162 160 158 157</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0</td>
<td>160 161 162 162 162 161 159 158</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0</td>
<td>158 159 161 161 162 162 159 158</td>
</tr>
<tr>
<td>Inverse Quantized Coefficients</td>
<td>Reconstructed Image Sub-block</td>
</tr>
<tr>
<td>Figure 2.3 Subjecting an Image Block to JPEG / Inverse JPEG Processes</td>
<td></td>
</tr>
</tbody>
</table>
2.3.3 Entropy Coding of Quantized DCT Coefficients [141-143]

Prior to entropy coding, there usually are few nonzero and many zero-valued coefficients (see matrix in Figure 2.3(d)). The task of entropy coding is to encode these few coefficients efficiently. For ease of presentation the entropy coding procedure is described in several closely related steps as follows:

The first deals with the coding of DC coefficients. After quantization, the DC coefficient is treated separately from the \((N \times N - 1)\) AC coefficients. The DC coefficient is a measure of the average value of the \(N \times N\) image samples. As the DC coefficients of the adjacent \(N \times N\) blocks are strongly correlated, they are encoded as the difference from the DC term of the previous block in the encoding order (i.e. raster scan). This special treatment is worthwhile, as DC coefficients frequently contain a significant fraction of the total image energy.

After converting the DC element to a difference from the last block, the \(N \times N\) coefficients are arranged in the zigzag order as illustrated in Figure 2.2. This ordering helps facilitate entropy coding by placing low-frequency coefficients (which are more likely to be non-zero) before high-frequency coefficients.

The description of entropy coding is introduced in two steps: conversion of the DCT coefficients into an intermediate sequence of symbols and assignment of variable-length codes to the symbols. [141].

In the intermediate symbol sequence, each nonzero AC coefficient is represented in combination with the "runlength" (consecutive number) of zero-valued AC coefficients that precede it in the zigzag sequence. Each such runlength / non-zero coefficient combination is usually represented by a pair of symbols: Symbol-1 (runlength, size) and Symbol-2 (amplitude). *Amplitude* is simply the amplitude of the nonzero AC coefficient. *Runlength* is the number of consecutive zero-valued AC coefficients in the zigzag sequence preceding the nonzero AC coefficient being represented. *Size* is the number of bits used to encode the
amplitude, i.e., to encode symbol-2 by the signed-integer encoding used with the particular method of Huffman coding.

This process can be explained in relation to the intermediate sequence of symbols of the image sub-block given in Figure 2.3 (a). When the AC coefficients are ordered in a zigzag format, the first non-zero coefficient is \(-2\) (see Figure 2.3(d)). This is preceded by a single zero. Thus, the intermediate representation is written as \((1,2)(-2)\) (runlength=1, size=2, amplitude = -2). Next encountered are three consecutive non-zeros of amplitude \(-1\), giving three consecutive \((0,1)(-1)\) intermediate symbol representations. The last non-zero coefficient is \(-1\) preceded by two zeros, which gives a representation of \((2,1)(-1)\).

Runlength represents zero-runs of length 0 to 15. Actual zero-runs in the zigzag sequence can be greater than 15, so the symbol-1 value \((15,0)\) is interpreted as the extension symbol with runlength=16 [141]. There can be up to three consecutive \((15,0)\) extensions before the terminating symbol-1 whose runlength value completes the actual runlength. The terminating symbol-1 is always followed by a single symbol-2, except for the case in which the last run of zeros includes the last \((N\times N-1)\) AC coefficients. In this frequent case, the special symbol-1 value \((0,0)\) means EOB (end of block), and can be viewed as an escape symbol that terminates the \(N\times N\) sample block.

The intermediate representation for the \(N\times N\) sample block’s differential DC coefficient is structured similarly. Symbol-1 however represents only size information; symbol-2 represents amplitude information as before. If the DC coefficient of the preceding block was assumed to be 12, which results in a differential DC value of 3, the intermediate representation for the DC coefficient of the image sub-block of Figure 2.3 (a) would be \((2)(3)\).

Thus, the intermediate sequence representation for the image sub-block is,

\[
(2)(3), (1,2)(-2), (0,1)(-1), (0,1)(-1), (0,1)(-1), (2,1)(-1), (0,0)
\]
Once the quantized coefficient data for a $N \times N$ block is represented in the intermediate symbol sequence described above, *variable-length codes* are assigned. For each $N \times N$ block, the DC coefficient's symbol-1 and symbol-2 representation is coded and output first. For both DC and AC coefficients, each symbol-1 is encoded with a *variable-length code (VLC)* from the Huffman table assigned to the $N \times N$ block's image component. Each symbol-2 is encoded with a *variable-length integer (VLI)*. VLCs and VLIs are both variable-length codes, but VLIs are not Huffman codes [197]. An import distinction is that the length of a VLC is not known until it is decoded, but the length of a VLI is stored in its preceding VLC. Huffman codes (VLCs) must be specified externally as an input to JPEG encoders. However, VLI codes in contrast, are hardwired into the proposal [142,143].

The differential DC VLC for the above example (i.e. for the starting symbol (2) in the intermediate sequence representation, above) is 011. The AC luminance VLCs for this example are:

- $(0,0)$: 1010
- $(0,1)$: 00
- $(1,2)$: 11011
- $(2,1)$: 11100

The VLIs specified in the standards are,

- $(3)$: 11
- $(-2)$: 01
- $(-1)$: 0

Thus the bit stream for the above example block is,

011 11, 11011 01, 00 0, 00 0, 00 0, 11100 0, 1010

Note that 31 bits are required to represent 64 coefficients, which achieves compression of just under 0.5 bits/sub-block.
The decoding process is basically the inverse of the processes discussed above and is illustrated in Figure 2.4.

### 2.4 Stereo Image Compression

Stereoscopic imaging systems, which are increasingly used for multimedia, tele presence, or telerobotics, permits us to enhance tridimensional perception by presenting a slightly different viewpoint of the scene to each eye. These systems have been shown to be beneficial for accomplishment of a number of tasks, such as object tracking [4], object manipulation [2,3,13,26] and relative depth perception [12,134]. Since the usage of stereoscopic systems imply a significant increase in the amount of visual information for remote transmission, it is essential to make use of compact representations of this information. The representations should, on one hand, take advantage of all the redundancy sources present in stereoscopic signals (Section 2.4.1), and on the other hand, only exploit the information, which is strictly necessary to the human observer (Section 2.4.2).

### 2.4.1 Characteristics of Stereoscopic Signals

Stereoscopic signals possess an additional source of redundancy (as compared to intra-image redundancy of still images and inter-frame redundancy of image sequences) created by the
observation of the same scene from two slightly different viewpoints. Most parts of a scene
are thus visible from the two viewpoints, and project on to similar regions of the left and
right image planes. In theory, this geometrical redundancy can be exploited for compression
purposes. A possible strategy consists of putting into correspondence regions of pre­
established size and shape (blocks, objects, regions) of a stereoscopic pair, and transmitting
disparity vectors (as compared to motion vectors in image sequences) allowing the
reconstruction of one viewpoint from the other. This prediction strategy is referred to as
disparity compensation.

2.4.2 Characteristics of Stereoscopic Perception

Characteristics and limitations of the human binocular perception may also be taken into
account to compress the representations for stereoscopic information beyond the limits
imposed by signal processing. For instance, depth resolution limits of the human visual
system can be considered in order to limit disparity vector amplitudes to values greater than
the depth discrimination threshold [123]. Some stereoscopic image representation systems
may not reproduce the full disparity range perceivable by the human visual system [123].
The fact that the depth resolution of the human visual system decreases exponentially with an
increase in distance from the fixation plane can also be exploited to represent disparity
vectors with a non-uniform resolution according to an exponential quantization function
[100]. Psychophysical data reported by A.Ariditi [95] show that the threshold of stereoscopic
perception as a function of increasing angular separation between target and a fixation
reference increases according to an exponential function, with maximal stereoacuity at the
fovea and a more rapid stereoacuity decrease after six degrees of visual angle [95]. Spatial
acuity is also maximal at the fovea, and decreases as a monotonic function of the distance of
target from fixation, because of the decreasing density of cone receptors [181]. These factors
can be taken into account to reduce the level of details in peripheral regions. [182].

The two main theories related to human stereoscopic vision, the suppression theory and the
fusion theory [95], also play an important role in the elaboration of stereoscopic coding
schemes.
2.4.2.1 The Suppression Theory

The subjective three-dimensional percept resulting from a stereo image pair with a degraded image, possesses a quality level approaching the one of the non-degraded image [95].

The suppression theory can be considered as the basis of asymmetrical coding schemes, where two images of a stereoscopic pair are treated differently, with one channel generally used as a reference for the prediction of the other channel. Asymmetrical coding schemes often take for granted the degradations resulting from the high compression of one of the two images of the pair, then are only slightly perceivable because the other image of the same pair keeps a good quality level [17,85].

2.4.2.2 The Fusion Theory

The two images of a stereoscopic pair are fused to give rise to a volumetric representation of the environment [95].

This theory motivates symmetrical coding schemes where an intermediate representation integrating information from the two stereoscopic channels is created. With symmetrical approaches, reconstruction errors are generally uniformly distributed between the two viewpoints. The intermediate representation can take different forms, such as another image obtained by combining the two original images or a 3-D model integrating information from the two viewpoints.

2.4.3 Theory of Stereo Image Coding

This section investigates the theoretical basis of coding correlated sources (stereo pair sources in particular) [139]. A stereo pair (X, Y) can be modelled as a vector-valued outcome of two correlated discrete random processes. First, let us consider the lossless encoding of a stereo pair. According to Shannon's theorem, a rate that is greater than $H(X,Y)$ (note: $H$-entropy), which is equal to $H(X) + H(Y|X)$ (note: $Y|X$ - conditional probability of $X$ given $Y$), is sufficient to recover the stereo source with arbitrarily low probability of error if they
are encoded together. An optimal lossless coding strategy, which is referred to as the conditional coder (CONCOD) for a stereo pair is, then, “to code one image and then code the second image given the coded first image”. On the other hand, another optimal coding scheme, known as the Slepian-Wolf coder, may be employed such that two sources are coded independently of each other and decoded by a common decoder [217]. The basic theorem of the Slepian-Wolf encoder is given as follows,

Slepian-Wolf Theorem [79]: Correlated sources X and Y can be separately described at rate $R_x$ and $R_y$ and recovered with arbitrarily low probability of error by a common decoder, if and only if,

$$R_x > H(X|Y)$$
$$R_y > H(Y|X)$$
$$R_x + R_y > H(X,Y)$$ \tag{2.8}$$

An important question, however, is how to design an optimum lossy coder, in the rate-distortion sense, for correlated sources. Specifically, “What is the minimum number of bits necessary to code the stereo source $(X,Y)$ such that the distortion for $X$ is $D_x$ and the distortion for $Y$ is $D_y$?” One possible solution is to use the CONCOD structure, which has been shown to be sub-optimal for some region in rate-distortion surface. Another candidate is the Slepian-Wolf coder. However, the general rate-distortion region remains unknown for the Slepian-Wolf problem with distortion in both $X$ and $Y$ [77]. It is useful to examine the performance of the CONCOD structure with respect to the optimal coder. It can be shown that [80],

$$R_x(D_x) \leq R_{x,y}(D_x,D_y) \leq R_x(D_x) + R_{y|x}(D_y)$$ \tag{2.9}$$

where, $R_x(D_x)$ is the rate-distortion function for $X$, $R_{x,y}(D_x,D_y)$ is the joint rate-distortion function for $X$ and $Y$ (the optimal coder), and $R_{y|x}(D_y)$ is the conditional rate distortion function for $Y$ given the encoded $X$ (the optimal conditional coder). As the percentage of the
total bit rate allocated to the second image (predicted image) is small, let us consider the extreme case, in which we allocate zero bits for the second image [the distortion $D_y$ for the second image is equal to its maximum value $D_{y_{\text{max}}}$, i.e., $R_{y|x}(D_{y_{\text{max}}}) = 0$]. For this case, the CONCOD structure is optimal, since the lower and upper bounds of (2.9) are identical. As a result of the continuity of the rate distortion functions, for any $\varepsilon > 0$ there exists a $\delta$ such that if $|D_{y_{\text{max}}}-D_y| < \delta$ then,

$$\left[R_x(D_x)+R_{y|x}(D_y)\right]- R_x(D_x)+R_{y|x}(D_{y_{\text{max}}}) = R_{y|x}(D_{y_{\text{max}}}) < \varepsilon$$

(2.10)

However, since,

$$R_{x,y}(D_x,D_y) = R_{x}(D_x)+R_{y|x}(D_{y_{\text{max}}})$$

(2.11)

Rewriting the left-hand side of (2.11) using (2.9) and substituting it into (2.10) yields the following inequality:

$$\left|R_x(D_x)+R_{y|x}(D_y)- R_x(D_x,D_y)\right| < R_{y|x}(D_{y_{\text{max}}}) < \varepsilon$$

(2.12)

Therefore, the CONCOD structure is performing arbitrarily close to the optimal solution given that $R_{y|x}(D_y)$ is small. In addition, conditional coders are backward compatible with monocular image coding systems. In the next subsection and subsequent chapters (3 & 4), several algorithms that have been proposed based on this observation, are discussed. For most coding systems, the human observer should be able to perceive the 3D scene when the coded stereo pair is presented to him/her. Therefore, instead of the distortion pair $(D_x, D_y)$, one may want to use the distortion $D_{sp} = D_x + D_y$ that quantifies the distortion in depth perception. However, no such distortion measure has been developed as yet.
2.4.4 Stereoscopic Image Capture

Stereo image pairs can be captured using two monocular camera’s rigidly mounted according to two camera configurations namely, parallel axis camera geometry and convergent axis camera geometry. In Section 1.5 it was mentioned that parallel axis camera geometry has been assumed in all stereo image compression algorithm designs and their testing, in this thesis. The following section gives an account of this camera geometry and its associated properties and compares them with the properties of its alternative, convergent axis camera geometry.

2.4.4.1 Parallel Axis Camera Geometry

Let us assume that we have two pinhole cameras whose optical axes are parallel and are separated by a distance w. The cameras each have focal length $l$, with $f_l$ the focal point of the left camera, and $f_r$ the focal point of the right. Let $f$ be the focal point of an imaginary cyclopean camera placed half way along the baseline, i.e. the line connecting the left and the right focal points. Assume that the cameras are placed so that the baseline is parallel to the image planes and perpendicular to the optical axes [1,26], (see Figure 2.5).

A point $p$ on the surface of an object in 3-D space (visible to all three cameras) is projected through the focal points and on to the image planes. Each image plane has a 2-D co-ordinate system with its origin determined by the intersection of its optical axis with the image plane. The brightness of each point projected onto the image planes creates image luminance functions $I_l$, $I_r$, and $I$ in the left, right and cyclopean planes, respectively.

A horizontal plane through the baseline intersects the three image planes in what are called epipolar lines, which are denoted by $X_l$, $X_r$ and $X$, with co-ordinates $x_l \in X_l$, $x_r \in X_r$, and $x \in X$, respectively. The co-ordinates of the epipolar lines run right to left, so that when a point moves from left to right, its co-ordinates in the image planes increase.
When the same point is visible from all three eyes it is easy to check that \( x = (x_l + x_r) / 2 \).

Thus, the co-ordinates of the points projected on to all three image planes can be related by a positive disparity function \( f(x) \) via, \( x_l = x + d(x) \) and \( x_r = x - d(x) \).

With this symmetric definition for the disparity, the following relationship holds.

\[
d(x) = \frac{x_l - x_r}{2}
\]  

(2.13)

If \( z(x) \) is the perpendicular distance from a line connecting the focal points to the point \( p \) on the surface of the object, then the disparity \( d(x) \) can be related to the distance \( z(x) \) by,

\[
d(x) = \frac{lw}{2z(x)}
\]  

(2.14)

Given that \( l \) and \( w \) are constants the above equation shows that the disparity \( d(x) \) is inversely proportional to the distance of the object from the image plane.
Chapter 2 Fundamental Concepts and Literature Review

The above simple mathematical relationship (i.e. equation 2.14) further infers that in parallel axis camera geometry the disparities could only be in the horizontal direction. This makes searching for correspondence points easier and less computationally intensive. However, this condition does not hold true in convergent axis camera geometry [1,21] where the cameras are placed such that their axes converge at a point. The parallel axis geometry also makes it easier to construct viewers for stereopairs acquired using this geometry. It is also easier to obtain and maintain this geometry, as the calibration becomes practically easier.

Although the proposed algorithms have been designed and implemented for parallel axis camera geometry they can be easily extended to meet the needs of convergent axis camera geometry. One straightforward change is in the shape of the search window, which would then have to be extended in the vertical direction as well.

2.4.5 Stereoscopic Image Display Technology

Stereoscopic display units can be broadly categorised as Head Mounted Displays (HMDs) and Off-Head Displays (OHDs).

The HMDs generally offer larger fields of view and the possibility of total immersion into the application space, which may be desired in certain applications (visualisation, VR games). However they suffer from weight concerns, resolution problems, computing power requirements and user discomfort (headaches and nausea).

The OHDs currently in use usually require that the devices for the image separation be worn by the observer. Shutter glasses in combination with time sequential presentation of the left and right eye images or polarisation glasses in combination with orthogonally polarised images (displayed either constantly by twin displays or time-sequentially by means of electro-optical shutters) mark the state of art in this technology.

Recently a number of free-viewing 3-D displays (autostereoscopic) based on the concept of direction multiplexing have been proposed [16,166,168]. Direction multiplexing (by optical means) results in different perspective views being visible only from specific locations.
Chapter 2 Fundamental Concepts and Literature Review

When the user positions his or her head so that the eyes are at the prescribed locations, both views are immediately fused to create the illusion of a three-dimensional space. These display concepts rely on optical effects such as occlusion (by parallax barriers, parallax illumination, moving slits), refraction (by lenticular plates, field lenses), reflection (by autocollimation screens), deflection (by prism masks) and diffraction (by holography, holographic optical elements [168], diffraction optical elements). At present, the autostereoscopic techniques are not as mature as the techniques requiring glasses.

2.4.6 A Comparison of Stereo Pairs and Video Sequences

Disparity and motion compensation are closely related. In fact, previously proposed disparity compensation techniques are generally adapted from approaches that are used for motion compensation in video coding. Stereo imaging is generally considered to be an optional extension to a basic monocular system. Thus, we should allow only a small distortion in the quality of the mono image, i.e. only a small percentage of the overall bit rate should be allocated for the second image. The problem of coding the second image of a stereo pair given the first appears to be similar to the problem of video coding, i.e. to encode the P frame given the I frame. However, important differences do exist between the two problems [128].

For example, for typical video sequences, where low bit rate coding algorithms are employed, background objects do not generally move from one frame to the next, and only a small percentage of the scene undergoes motion. The displacement for the moving objects, which may be modelled as purely translational, usually does not exceed a few pixels [129]. In fact, motion compensated video coding algorithms exploit these properties for higher compression rates [130]. By contrast, every object in a stereo pair is displaced and the displacements may be very large compared to video sequences [80]. As a consequence, the performance of disparity compensation for stereo images are generally lower than the performance of motion compensation for interframe video coding, especially for very low bit rates [121,128].

Standard block matching algorithms such as MSE (Mean Square Error) or MSD (Mean Square Distance), assume the axiom of constant intensity along the displacement trajectory.
For a video sequence, this axiom states that object points maintain the same luminance values from one frame to the next. For a stereo pair, the axiom states that the recorded intensity values by the right and left cameras are identical for all object points. While this axiom is generally valid for video sequences, it is rarely true for stereo pairs. The light intensity that is reflected from objects in a scene and recorded by the camera depends on the position of the camera relative to the scene [131]. This is a primary source of photometric variations. Other variations, however, may arise from noise and different photometric characteristics of the cameras. In fact, the corresponding regions may have different intensity values, and area-based disparity estimation may yield inaccurate correspondences. Histogram modification [121], global calibration [132], and normalised similarity measures [133] have been proposed to solve the false estimation problem in the presence of photometric variations. The first two techniques are global approaches and cannot handle local variations, whereas the third method is computationally intensive.

The third difference between a stereo pair and a video sequence is the source occlusion. In video sequences, occlusion occurs due to moving objects, as in the case of background covering. For stereo pairs, occlusion occurs when some part of the scene can only be captured by one of the cameras, due to the finite viewing area, which is referred to as the framing error, or due to depth discontinuity [80]. In some applications, by suitable horizontal cropping of the images, effects of occlusion due to finite viewing area can be minimised.

2.4.7 Review of Stereoscopic Image Compression Methods

Among the various compression methods for stereoscopic images, the simplest one consists of the independent coding of the left and right channels. However, this approach is far from efficient since the inherent stereoscopic redundancy is not exploited.

The first proposed stereo image compression algorithm (M.G.Perkins [122]) was to code the sum and difference of the two images. The theoretical reasoning behind this approach is that the sum and difference images obtained from a stereo image pair are non correlated if the right and left images have the same first and second order statistics. Three-dimensional discrete cosine transform coding of stereo pairs (I.Dinstein et al [121]) is equivalent to the
sum-difference coding in the transform domain. However, the performance of these technique decreases with increased disparity values.

A modification to the sum-difference coding is to shift one of the images horizontally to the point where the cross correlation between the images of stereo pair reaches its maximum value (H.Yamaguchi [123]) (the shift approximately equals the mean disparity of the scene). The shifted image is then subtracted from its partner image to remove the redundancy, and the difference is encoded [123]. This method, referred to as the global translation method, assumes that objects in the scene have similar disparity values. However, since objects in the scene have generally different disparity values, this method is not particularly efficient. Another approach is to translate the row blocks instead of the whole image. The amount of translation is determined either by cross correlation statistics (correlation enhanced technique) [123] or using the most interesting feature in the row block (E.Salari [124]).

M.G.Perkins [17] (mixed resolution coding) and I.Dinstein et al [163] have also proposed asymmetrical approaches that exploit the suppression theory by subsampling one of the two images. Even though these methods are attractive (as they do not require matching), they are not appropriate for applications requiring fine details perception [17], such as telemanipulation for example.

Disparity compensation represents another example of asymmetrical coding [17,21,125]. Lukacs [125] was the first to introduce disparity compensated prediction. The aim of disparity compensation is to estimate the disparities between the objects in a stereo pair, and use these estimates to remove the stereo redundancy. The idea is to use a disparity corrected version of one image as a predictor of its partner image. Since the disparity information should be transmitted, in applications where the compressed stereo pairs will be viewed by a human, computing a sparse disparity field is the key to compression [10].

The disparity is usually calculated via a block-based scenario. First, the left image is independently coded. Then the right image is divided into non-overlapping blocks. Either fixed or variable size blocks are used. Each of these blocks is shifted horizontally and compared to the corresponding blocks in the coded left image using some (MSE, SAD, etc.)

32
measure to determine the similarity between the two blocks. The most similar block, is the 
disparity compensated prediction, and the corresponding translation is the disparity for the 
block.

Given the right image that is to be encoded and a disparity vector field, there are a variety of 
coding strategies that may be used. For example, the residual of the disparity compensated 
prediction may be encoded and transmitted. This method is referred to as Disparity 
Compensated Residual Coding (DCRC) and was first proposed by H.Yamaguchi et al 
[123,126].

In an alternative approach to encoding the residual, proposed by Dinstein et al [121], each 
block is either encoded using disparity compensated prediction (only the block disparity for 
the block is transmitted), or the block is independently coded using an adaptive DCT. The 
approach used for a given block is determined by the accuracy of the disparity compensated 
prediction.

M.G.Perkins [17] proposed a Disparity Compensated Transform Domain Predictive 
(DCTDP) coder that estimates the transform of the right image blocks, \( R = T(b_r) \), from the 
transform of the matching left image block, \( L = T(b_l) \), using \( \hat{R} = A \otimes L + B \), where the array 
multiplication operation \( \otimes \) is performed by multiplying the \((i,j)\)th element of the first 
matrix by the \((i,j)\)th element of the second matrix. The matrices \( A \) and \( B \) are chosen based on 
the statistics obtained from a training set so that each coefficient is predicted using a 
minimum variance predictor. After estimating the transform domain coefficients, the 
transform domain error is quantized and sent through the channel. The minimum variance 
predictor performs better than the simple predictor proposed by H.Yamaguchi et al 
[123,126], only for bit rates higher than 0.5 bits/pixel [17]. This method is discussed in detail 
in Sections 2.5 and 3.1, and used as the benchmark algorithm (see Section 1.5) to test the 
new techniques proposed in this thesis.

As mentioned previously, disparity compensation can be implemented using both fixed block 
size or variable block sizes. Fixed block size schemes have the unique advantage of
eliminating the overhead that is required to specify the block locations. However, if multiple objects or an occluded object exist in a block, these schemes cannot fully exploit the stereo redundancy. One solution is to decrease the block size, which will increase the overall bit rate for the disparity field. Another approach is to use a segmentation approach. However, not all segmentation techniques are suitable due to the excessive number of bits required to describe the shape and the location of each region. Quadtree decomposition provides a relatively economical and effective solution to this problem. The idea is to decrease the block size adaptively when the prediction fails. V.Seferidis et al [127] and H.Aydinoglu et al [76] have implemented a regular quadtree approach, where a block can be divided only at the midpoint of its sides. Also, a generalised quadtree can be considered where a block can be divided at $2^k - 1$ locations ($k$ is the number of bits that can be allocated per side per node) (S.Sethuraman et al [53]). The division takes place as a permitted location that lies closest to a sharp intensity discontinuity. In general, variable block size schemes outperform the fixed block size schemes.

Block matching cannot handle non-linear deformations such as perspective distortion. Perspective distortion occurs when the object surface is not perpendicular to the camera axis. A generalised block matching algorithm that approximates the deformations of the objects, by deforming the corresponding blocks in the picture, has been proposed by V.Seferidis et al [127] to overcome this problem. This algorithm is used in conjunction with the quadtree decomposition. However, perspective distortion is not severe for outdoor images and can sometimes be properly handled if we only employ a quadtree decomposition.

The conventional disparity compensation algorithms using rectangular blocks often give discontinuities between neighbouring disparity compensated blocks in the predicted images. A windowed disparity compensation (WDC) scheme has been proposed by H.Aydinoglu et al [76] to overcome this problem.

Other geometric relations than the ones concerning disparity can also be exploited. For instance, information about projective transformations (L.Oisel et al [162]) to reconstruct one viewpoint from the other can also be transmitted. H.Aydinoglu, M.H.Hayes III, have proposed a method [137] that is based on disparity compensation and subspace projection.
The *subspace projection technique* (SPT) is a transform domain approach with a space-varying transformation matrix and may be interpreted as a spatial transform domain representation of the stereo data. One advantage of this approach is that it can locally adapt to the changes in the cross-correlation characteristics of the stereo pairs.

*Symmetrical approaches* that involve the creation of an intermediate representation can also be considered. For instance, an intermediate image corresponding to the image that would be seen from a camera positioned at the centre of the stereoscopic system can be obtained from the disparity field between the two original stereoscopic images. Intensity values in this image are obtained by averaging the intensity of the corresponding pixels in the left and right images (D.De Vleeschauwer [117]).

A stereo image pair can be coded using Visual Pattern Image Coding (VPIC) due to the capability of its coding primitives to reflect meaningful physical properties of projected real-scene surfaces: high-information edge regions and uniform regions. An algorithm for spatial matching of VPIC primitives has been proposed by D.Craievich and A.C.Bovik [39]. A dense disparity map is obtained my matching these primitives and used together with one of the VPIC coded images of the stereo pair to predict the second image.

A Wavelet Transform based method, which is based on the suppression theory of binocular vision, has been proposed for stereo image compression by W.D.Reynolds Jr., and R.V.Kenyon [40]. This is a frequency domain approach where the wavelet transform has been used to obtain a multi-resolution analysis of the stereo pair. The resulting sub bands convey the necessary frequency domain information, enabling the suppression theory to be used in the compression.

Vergence movements and higher resolution around the fovea (point of attention) are characteristics of the human visual system. It has been shown that spatially varying distribution of pixels is useful for better depth perception with stereo. In their research work A.Basu, W.Wong and J.Baldwin [41], discuss how existing stereo image compression algorithms can be improved by incorporating vergence and spatially varying sensing.
M.Forman, A.Aggoun and M.McCormick [15], have proposed a full parallax 3D image compression algorithm that takes advantage of the cross correlation between the multiple images recorded on a charge coupled device (CCD) placed behind a directionally selective micro-lens array as well as the correlation inherent within each image. A hybrid DPCM/DCT coding scheme is used to take advantage of the redundancies present within each sub-image and between neighbouring sub-images.

The two most common methods of acquiring 3D data are stereo techniques and the laser range finder. A stereo image matching algorithm directed by range images is proposed by Y.G.Wu, J.Yang, K.Liu and J.Y.Yang [134], from the point of view of sensor fusion. At first, the transformation between the range images and camera images is built up and then the information extracted from the range images is used to constrain the search in stereo matching since the computation of 3D feature points is fast in it. As a result the workload of the point correspondence in stereo can be significantly reduced.

S.Randriamasy and A.Gagalowicz [28], have proposed some algorithms developed to investigate the problems of region-based segmentation and matching processes on monochrome and colour stereo image pairs. A co-operative automatic top down segmentation algorithm is proposed, where two segmentation trees are constructed progressively. Regions are validated if a satisfying match is found, and then their division is stopped in the trees. The algorithm is data driven and produces usable output for the matching and 3D reconstruction processes even from the first division steps.

D.Q.Huynh and R.A.Owens [20] have proposed an algorithm for extracting regions and subsequently labelling line segments in images and then using the scheme for stereo matching. The focus of their work is on the description of an algorithm that uses the geometrical layout of line segments in an image to attach a label to each line segment: boundary, join or isolated. This line labelling is then used in conjunction with the planar regions within each image of a stereo pair to facilitate high level stereo matching.

H.Aydinoglu, F.Kossentini, Q.Jiang and M.H.Hayes III [43], have proposed a region-based algorithm for stereo image coding. Three types of regions are considered: occlusion, edge
and smooth regions. The region in the right image that is occluded due to finite viewing area is independently coded. The non-occluded region is segmented into edge and smooth regions, where each region is composed of fixed size blocks. The disparity for each block in a non-occluded region is estimated using a block-based approach. The estimated disparity field is encoded employing a lossy residual uniform scalar quantizer and an adaptive arithmetic coder based on the segmentation information. The coded vectors are used for the subspace projection technique, which is a combined disparity and illumination compensation algorithm.

Traditional stereo image coding techniques employing block based disparity compensation have problems with occlusion regions and photometric variations. H.Aydinoglu and M.H.Hayes III [44] have proposed an approach to stereo image coding based on disparity compensation and subspace projection. It has been shown that the above negative effects can be reduced if a Subspace Projection Technique (SPT) is used as a post processing method after disparity estimation. SPT is an orthogonal basis approach, which combines block-based disparity estimation with two-dimensional first order approximation. For regions where disparity estimation fails, such as occlusion regions, first order approximation provides a decent estimate. On the other hand, zeroth order term compensates for photometric variations.

2.4.8 Stereoscopic Image Sequence Compression.

For the case of stereoscopic sequences, many redundancy sources are simultaneously available, including intra-image structure, motion, and stereoscopy. Various methods, inspired from the MPEG compression standard, have been proposed to exploit stereoscopic redundancy.

A selection module (for selecting the coding mode) is generally used for encoding image blocks by considering the reconstruction errors at the receiver. The left channel is coded by means of a DCT with motion compensation. The right channel is coded by motion compensation, disparity compensation, or direct block encoding with a DCT, choosing for each block the estimate yielding the smallest reconstruction error (A.Kopernik [115]).
Disparity or motion compensation can also be applied to an intermediate image obtained by multiplexing the left and the shifted right viewpoint (F.Chassaing et al [116]).

In order to avoid artefacts associated with block-based approaches when high compression rates are applied, it is also possible to treat the image information in terms of two-dimensional objects (M.Ziegler [114]).

An iterative procedure is to segment the images into uniform regions, corresponding to objects, using available stereoscopic information. Motion information is also taken into account during the segmentation. Moving objects are then transmitted with a set of parameters describing their colour, form, motion, and disparity. In regions that cannot be modelled satisfactorily, the temporal difference between the successive images is transmitted in a conventional manner. Stereoscopic and dynamic information can also be integrated into a common and compact description of the scene, consisting of a 3-D structure and 3-D motion parameters (J.L.Dugelay et al [33], N.Grammalidis et al [34]).

Object modelling can facilitate motion and disparity estimation by taking into account, for instance, the difference between the current image and the projection of the estimated 3D model (L.Falkenhagen [113]). The transmitted information can be represented in terms of 3-D objects composed of planar surfaces, with motion described by rotation and a translation component (J.L.Dugelay et al [33]). The 3D surface can also be represented by depth information at zero crossings, which have been obtained from a contour detector (H.Morikawa [108]). Each object can be represented by 3D motion, it’s 3D structure, and it’s texture parameters (L.Falkenhagen [113]), with the object structure representation consisting of a 3D triangular mesh.

Even though 3D scene modelling represents an attractive coding approach for stereoscopic sequences, the complexity of this modelling is often very important. It is also possible that this modelling be inadequate in some image regions because of occlusions, for instance. One advantage of the explicit extraction of depth information is that frontal objects can easily be separated from the background (M.Waldowski [107]). It then becomes possible to treat these two entities differently, using hybrid approaches combining 2D and 3D representations. The
selection of an object mode (3D) or a block mode (2D) can be done with the goal of minimising reconstruction errors (N.Grammalidis [101]) or by taking into account the magnitude of depth variations (A.Kopernik [115]).

M.W.Siegel, P.Gunatilake, S.Sethuraman and A.J.Jordan [8], have proposed an algorithm that exploits the high correlation in stereoscopic image sequences by straightforward computing of blockwise cross-correlation’s and multi-resolution hierarchical matching using a wavelet-based compression method.

S.Sethuraman, M.W.Siegel and A.G.Jordan [9], have used the suppression theory for stereoscopic sequence compression. One of the bit streams is independently coded along the lines of MPEG 2 standard, while the other stream is estimated at a lower resolution from this stream. A multi-resolution framework is adapted to facilitate such an estimation of motion and disparity vectors at different resolutions.

Stereoscopic image sequence transmission over existing monocular digital transmission channels without seriously affecting the quality of one of the image streams, requires a very low bit rate coding of the additional stream. Fixed block-size based disparity estimation schemes cannot achieve such low bit-rates without causing severe edge artefacts. Textureless regions lead to spurious matches that hamper the efficient coding of block disparities. S.Sethuraman, M.W.Siegel and A.G.Jordan [49], have proposed a novel disparity-based segmentation approach, to achieve an efficient partition of the image into regions of more or less fixed disparity. The partitions are edge based, in order to minimise the edge artefacts after disparity compensation. The scheme leads to disparity discontinuity preserving, yet smoother and more accurate disparity fields than fixed block size based schemes. The smoothness and the reduced number of block disparities lead to efficient coding of one image of a stereo pair given the other. The segmentation is achieved by performing a quad-tree decomposition, with the disparity compensated error as the splitting criterion. The multi-resolution recursive decomposition offers a computationally efficient and non-iterative means of improving the disparity estimates while preserving the disparity discontinuities. The segmented regions can be tracked temporally to achieve very high compression ratios on a stereoscopic image stream.
The same authors have proposed an alternative method [52] for stereoscopic image sequence compression where each frame in one of the streams is segmented based on disparity. An MPEG-type frame structure is used for motion compensated prediction of the segments in this segmented stream. The corresponding segments in the other stream are encoded by reversing the disparity map obtained during the segmentation. Areas without correspondence in this stream, arising from binocular occlusions and disparity estimation errors, are filled in using a disparity map based predictive error concealment method.

D.V.Papadimitriou, T.J.Dennis [11], have shown that if a scene is captured in stereo, the analysis of moving image sequences for 3D modelling can be performed in a relatively straightforward manner. The output from a stereo disparity estimation process using calibrated cameras gives absolute 3D surface co-ordinates from a single stereo pair. When combined with monocular motion cues, the true 3D motion parameters of moving objects can be accurately calculated. Further analysis enables segmentation of body elements according to motion while the 3D surface feature structure, although available from the start, can be integrated and expected to alleviate the known problems of ambiguity suffered by monocular source model-based coders.

A Kalman Filter based algorithm for 3D motion estimation from a stereo image sequence using a unified temporal-spatial optical flow field (UOFF) has been proposed by J.N.Pan, Y.Q.Shi and C.Q.Shu [45].

B.Choquet, J.L.Dugelay and D.Pele [18] have focused their research on the algorithm aspects for defining an optimal coder well suited to 3DTV. A 3D approach, linking 2D left and right motion is presented in order to define a 3D-coding scheme based on motion estimation and compensation. This 3D scheme allows for the possibility of using a unique channel for transmitting motion estimation-compensation information instead of two or three channels.

F.Labonte, C.T.Le Dinh, J.Faubert and P.Cohen [139], have proposed a new compression scheme for interlaced stereoscopic sequences which differentiates between a region of fixation and a peripheral area, and thereby compacts the stereoscopic information into the spectral space of a monocular video channel. Spectral compression is achieved by avoiding
transmitting high frequency information over entire images, but only within and around the region where the observer acuity is the highest. The proposed approach consists of decomposing the left and right fields of the stereoscopic pairs into low-pass and high-pass components. High frequency components are then limited to a fixation region, thus allowing a reduction of their spectral extent. A composite video signal is then formed by positioning the different components into the available spectral space through filtering and modulation. Strategies have been proposed for the estimation of the fixation region, based upon a psychophysical study on visual strategies during depth discrimination tasks.

S. Malassiotis and M. G. Strintzis [94], have proposed an object-based stereo image coding algorithm that relies on the modelling of the object structure using 3D wire-frame models, and motion estimation using globally rigid and locally deformable motion models. Motion parameters are used to construct predicted images at subsequent time instances by mapping the image texture on the object surface. The authors also propose [19] a joint estimation of motion and disparity vector fields from stereoscopic image sequences by modelling the local interaction processes using Markov Random Fields (MRF). Interaction of the neighbouring motion disparity vectors across a discontinuity line is prohibited via hidden Markov fields, signalling discontinuities in the vector fields. Occlusion processes are used to mark occluded image locations, which may yield ambiguous matches. The coherence of motion and disparity vector fields is exploited by means of the epipolar constraint [42] (Corresponding points on a stereoscopic image pair are constrained to lie on the epipolar lines, i.e. the lines defined by the intersection of the epipolar plane and the image planes) and the so-called 'loop constraint' (mutual correspondence of four points in two subsequent stereo images). A simulated annealing algorithm is employed to find the global maximum of the posterior probability.

N. Grammalidis, S. Malassiotis, D. Tzovaras and M. G. Strintzis [135] have proposed another method that performs 3D-motion estimation for stereoscopic image sequences. The 2D motion of each object observed in one of the two views is modelled using a 3D-motion model involving a translation and a rotation. The estimation of the model parameters is performed in two steps: a linear step involving 2D vectors that are initially estimated using block matching techniques followed by a non-linear step involving displaced frame
difference minimisation. The regions where the 3D model is applied are identified using a motion based split and merge technique. Furthermore an extension of the 3D motion estimation method that uses a single 3D motion model to describe the apparent 2D motion in both channels has been proposed. These 3D motion estimation methods are then integrated in a stereoscopic inter-frame coding scheme. A hybrid coder, using block-based coding as a fallback mode in cases where 3D-motion estimation fails, has also been proposed.

An object-based coding scheme that is in many ways common with the above method has been proposed by D.Tzovaras, N.Grammalidis and M.G.Strintzis [136]. A multi-resolution block-based motion estimation approach is used for initialisation, while disparity estimation is performed using a pixel based hierarchical dynamic programming algorithm. A split and merge segmentation procedure based on both 3D motion and disparity is then used to determine regions with similar motion and depth parameters. This is combined with an efficient depth modelling method that offers full depth information at the decoder site. In order to reduce the computational load of merge phase of the algorithm, a fast algorithm is implemented which speeds up the merge procedure considerably. The segmentation part of the algorithm is interleaved with the estimation part in order to optimise the coding performance of the segmentation procedure. Motion and depth model parameters are then quantized and transmitted to the decoder along with the segmentation information. An object-based motion compensation scheme is then used to reconstruct the original image, based on the objects created by the segmentation approach.

A method to obtain an intermediate image sequence from a stereoscopic sequence consists of temporally multiplexing the left and the right image sequences in an adaptive manner. It takes into account the amplitude of reconstruction errors coming from motion or disparity compensated predictions. F.Chassaing et al [116] have proposed a scheme for stereoscopic image sequence compression using this approach.
2.5 Disparity Compensated Transform Domain Predictive Coding
- The Benchmark Algorithm -

This algorithm has been used as a benchmark to assess the performance of the stereo image compression algorithms proposed in this thesis. This pioneering algorithm of block based stereo image compression was introduced by M.P. Perkins [17] in 1991. The work has been widely quoted in subsequent research work and has been used as a benchmark to test most of the stereo image compression techniques introduced by various authors in the recent past. The algorithm generally works well for most types of stereo image pairs. Figure 2.6 shows the basic block diagram of the encoder.

![Block Diagram of the DCTDP Encoder](image)

**Figure 2.6 Block Diagram of the DCTDP Encoder**

**Basic Compression Strategy:** The left frame is first encoded using the JPEG lossy compression standard [142,143]. However, the right frame is encoded based on the left frame. It is first blocked into non-overlapping 8×8 sub-blocks, and these blocks are used to search the left frame for the best possible match within a specified window area (Figure 2.7). A suitable matching criteria such as, mean squared error (Section 2.6.2) or sum of absolute differences (Section 2.6.3), can be used. Once the best match is found the prediction error (in pixel domain) is calculated as the difference between the block to be encoded and the best match selected from the left frame. The prediction errors are then transformed into the
frequency domain using a DCT (Section 2.3.1) and quantized using scalar quantization (Section 2.3.2). The quantized coefficients are then entropy coded (Section 2.3.3) and transmitted as in JPEG. The displacement between the two blocks, i.e. block disparity, is coded and transmitted separately.

Figure 2.7 shows a schematic of the disparity compensation and prediction process.

![Disparity Compensated Predictive Coding Diagram](image)

The interframe redundancy between the two images would result in the encoded block finding a very good match from the left frame windowed search. This is the basic premise behind this compression algorithm. As the stereo image pair is assumed to be obtained using parallel axis camera geometry, the search is limited to the horizontal direction (see Section 2.4.5).

At the decoder end the disparity values are used to find the predictors for the blocks to be decoded and the corresponding error block is added to form the decoded block. The process is continued until all the blocks have been decoded.

The block matching procedure discussed above has only integer pixel accuracy. In other words, the matching block positions are found to the closest integer pixel location. It is a known fact in such a block matching procedure that the correlation between the low frequency coefficients of matching blocks is far greater than the correlation between the high
frequency coefficients. In order to address this issue M.G.Perkins suggested the use of a linear predictor [17]. Later we demonstrate (Section 3.1 of Chapter 3), how such a prediction scheme could improve the overall compression efficiency of the above mentioned algorithm. The effects of photometric variations due to the differences of camera positions and digital circuitry (gain etc.) of the two camera’s would also be somewhat compensated by the use of this linear predictor [17].

### 2.6 Important Definitions

This section defines several quality measures that have been used in the subsequent chapters.

#### 2.6.1 Extra Compression Ratio (ECR)

The *Extra Compression Ratio* is used as a measure to compare the performance of the proposed algorithms and the benchmark algorithm (*DCTDP*) [17] (Section 2.5) as against the direct compression of the predicted frame (right) using the lossy JPEG standard. Thus, ECR is defined as a percentage as follows.

\[
ECR = \frac{\text{bit}_\text{NoJPEG} - \text{bit}_\text{NoALGORITHM}}{\text{bit}_\text{NoJPEG}} \times 100\%
\]  

(2.15)

Where, \( \text{bit}_\text{NoJPEG} \) and \( \text{bit}_\text{NoALGORITHM} \) are respectively, the total number of bits produced when the right frame is JPEG coded and when it is coded using one of the algorithms.

#### 2.6.2 The Mean Squared Error (MSE)

The mean squared error between image blocks \( A = \{a_{ij}\} \) and \( B = \{b_{ij}\} \) of size \( n \times m \) is given by,

\[
MSE = \frac{\sum_{i=0}^{n-1} \sum_{j=0}^{m-1} (a_{ij} - b_{ij})^2}{n \times m}.
\]  

(2.16)
2.6.3 The Sum of Absolute Differences (SAD)

The mean squared error between two image blocks \(A = \{a_{ij}\}\) and \(B = \{b_{ij}\}\) of size \(n \times m\) is given,

\[
SAD = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} |a_{ij} - b_{ij}|
\]  

(2.17)

2.6.4 Peak Signal to Noise Ratio (PSNR)

Peak Signal to Noise Ratio is used as a measure of image quality. Let \(Rec(i,j)\) and \(Org(i,j)\) represent the reconstructed and original pixel values of a given image frame of size \(n \times m\), respectively. The PSNR (in dB) of the reconstructed image is defined as,

\[
PSNR = 10 \log_{10} \left[ 255^2 \left/ \left( \frac{\sum_{i=0}^{n-1} \sum_{j=0}^{m-1} (Rec(i,j) - Org(i,j))^2}{n \times m} \right) \right] \right]
\]  

(2.18)
Chapter 3

Block-Based Coding of Stereo Image Pairs

This chapter introduces the reader to the novel block-based strategies developed for stereo image pair compression. It starts with a review of the DCTDP algorithm proposed by M.G.Perkins [17] (see Section 3.1) who used a first order linear predictor to improve transform domain prediction in coding stereo image pairs. Section 3.2 includes the design, implementation and test details of the novel disparity field, coding scheme (One Bit Coder), that makes use of the high probability of two adjacent blocks having equal disparity values to obtain an overall reduction of the overhead bits. The novel Algorithm described in Section 3.3, completely eliminates the necessity of disparity field coding by using a pioneering block-based predictive coding scheme, to efficiently exploit both intra-frame and inter-frame correlation.

3.1 The Use of a First Order Linear Predictor to Improve Prediction

3.1.1 Introduction

Previous research has shown that in block matching with integer pixel accuracy, that the correlation between low frequency components of matching blocks is greater than the correlation between the high frequency components [17].

Thus, if $R$ is the 2D transform (frequency domain representation) of a given block in the right frame and $L$ is the 2D transform of its matching sub-block from the left frame a linear predictor of the form $\hat{R} = A \otimes L$, would significantly improve the compression performance over using the simple predictor $\hat{R} = L$. Here, $A \otimes L$ is the matrix formed by multiplying the $(i,j)$ th element of matrix $A$ (where, $1 \leq i, j \leq 8$), by the corresponding element of matrix $L$. 
In order to compensate for the differences in the digitisation circuitry of the two cameras a second parameter, matrix $B$ needs to be introduced to the above equation, i.e. the prediction equation should be changed to $\hat{R} = A \otimes L + B$. As the differences are mainly in the gains of the digitisation circuits, the matrix $B$ affects only the DC component of the transformed matrix. However, it is advisable to keep $B$ for all other frequency components as well in anticipation of unknown errors that may be present in the camera configuration.

Figure 3.1.1 Stereo Codec with First Order Linear Prediction
3.1.2 A and B, Parameter Calculation

The matrices $A$ and $B$ are obtained as follows. Each test image is divided into $8 \times 8$ sub-blocks. The $A$ and $B$ matrices for a particular stereo image pair is selected based on the statistics of the remaining stereo image pairs in the set of test images. Let ‘$m$’ be the number of sub-blocks in an image and ‘$n$’ be the number of test image pairs in the image set, excluding the one under consideration. Then, for every $(i, j)$ th element (where $1 \leq i, j \leq 8$), we have a set of $n \times m$ values for $R$ and $L$. $A$ and $B$ are found using curve fitting, i.e. by fitting a straight line, $\hat{R} = A \otimes L + B$ in the least mean squared error terms. Figure 3.1.1 above, shows a schematic that indicates the use of the linear predictor in the structure of the codec.

Once $A$ and $B$ are calculated for a particular stereo image pair using the above procedure, the prediction errors are calculated using $E = R - \hat{R} = R - (A \otimes L + B)$, for every matching block pair. These prediction errors are then quantized, coded and transmitted using JPEG. It has to be noted that the above prediction process is non-adaptive and that $A$ and $B$ matrices are the same for all matching blocks in every stereo image pair of the test image set.

3.1.3 Experimental Results & Analysis

To study the effects of using a linear predictor in block matching a set of five standard stereo image pairs derived from the web site, http://vision.stanford.edu/~birch/research/prp/ were used. All image pairs were of size $320 \times 320$ pixels. All were processed with a block size of $8 \times 8$. Table 3.1.1 below, shows the results obtained. Note that the ECR figures presented here only illustrate the extra data compression for encoding all right frames. For left frames the compression is entirely dependent on JPEG. The standard JPEG luminance quantisation table was used as the quantizer for all experiments.

It has been assumed that the disparity values of stereo image pairs will in general vary in between 0-63 pixels. This range was found to be adequate when the disparity values of a
large set of stereo image pairs were investigated. Note that the maximum disparity associated with a stereo image pair is dependent on how close, the closest object of the scene being viewed is from the camera image plane. In accordance with the above assumption, we have taken six bits as necessary, to encode each disparity value, under the conventional scheme described in Section 3.2.2.1

<table>
<thead>
<tr>
<th>Stereo Image Pair</th>
<th>Simple Predictor</th>
<th>Linear Predictor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cans (a1 &amp; a2)*</td>
<td>12</td>
<td>25</td>
</tr>
<tr>
<td>Stud. Lamp (b1 &amp; b2)</td>
<td>04</td>
<td>18</td>
</tr>
<tr>
<td>Lab (c1 &amp; c2)</td>
<td>21</td>
<td>35</td>
</tr>
<tr>
<td>Pack (f1 &amp; f2)</td>
<td>24</td>
<td>38</td>
</tr>
<tr>
<td>Slanted</td>
<td>11</td>
<td>27</td>
</tr>
</tbody>
</table>

Table 3.1.1 Experimental Results – Linear Prediction

The matrix $A$ derived from the above image data using the procedure that was discussed above is shown in Table 3.1.2.

<table>
<thead>
<tr>
<th>Table 3.1.2 Matrix $A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9986 0.9906 0.9448 0.9733 0.9243 0.8864 0.7520 0.6044</td>
</tr>
<tr>
<td>0.9955 0.9679 0.9438 0.9180 0.8586 0.8011 0.6274 0.5066</td>
</tr>
<tr>
<td>0.9895 0.9536 0.9341 0.9026 0.8503 0.6948 0.5926 0.5248</td>
</tr>
<tr>
<td>0.9799 0.9294 0.9037 0.8586 0.7602 0.7147 0.4853 0.4412</td>
</tr>
<tr>
<td>0.9648 0.9113 0.8641 0.8035 0.7214 0.5931 0.4736 0.4527</td>
</tr>
<tr>
<td>0.9601 0.8651 0.7992 0.7511 0.6524 0.5499 0.3937 0.4139</td>
</tr>
<tr>
<td>0.9444 0.8586 0.7559 0.7189 0.5994 0.4949 0.2758 0.2637</td>
</tr>
<tr>
<td>0.9439 0.7744 0.6384 0.5587 0.4344 0.3695 0.2045 0.2253</td>
</tr>
</tbody>
</table>
Figure 3.1.2, illustrates the reconstructed image quality of the stereo image pair, 'Lab'.

This is the reconstructed image obtained, when the linear predictor $\hat{R} = A \otimes L + B$ was used in the prediction process.

The results tabulated in Table 3.1.1 show that the linear predictor $\hat{R} = A \otimes L + B$ gives improved compression performance as compared to the simple predictor $\hat{R} = L$. The elements of matrix $A$ in Table 3.1.2 proves that the low frequency coefficients of matching sub-block pairs have much higher correlation (the lowest 15 frequency components have correlation values in excess of 0.9) than their high frequency
counterparts. (The highest 10 frequency components have correlation values less than 0.5). The matrix obtained for $B$ showed that only the element representing the DC component had a significant value (-0.3307). This is due to the gain differences of the digitisation circuitry of the two cameras. However, the rest of the elements in matrix $B$, although not so significant in magnitude, showed that there were some unknown errors associated with the camera circuitry and the configuration.

### 3.1.4 Conclusions

The above section has investigated and proved the known fact in block matching with integer pixel accuracy that, the low frequency components of matching sub-blocks have higher correlation as compared to the high frequency components. As a result, a linear predictor of the form $\hat{R} = A \otimes L + B$ can improve the compression performance when compared to the simple predictor $\hat{R} = L$. $A$ and $B$ values used in the above experiments were taken as fixed and were found by averaging over the whole set of images in the test image set.

Further experiments indicated that the elements of matrix $A$ depend on the local texture and the disparity value of the encoded block. This is an indication that a dynamic linear predictor that makes use of the local spatial redundancy would give an improved performance.

**An Important Note**

*As the linear predictor indicated a clear improvement in prediction, it was decided that this technique should be adapted in the prediction stage of all the algorithms that are designed, i.e., after disparity compensation of a block, $A$ and $B$ values are used in it's prediction. However, for clarity of presentation, this is avoided being repetitively mentioned in the authors contributed work, presented in Sections 3.2 & 3.3 and in Chapter 4.*
3.2 A Novel Coder for Disparity Field Coding

3.2.1 Introduction

In stereo image pairs, when objects move relative to each other due to binocular parallax, such objects may include a set of sub-blocks that move exactly by the same amount, resulting in an equal disparity value for all. This was observed to be particularly true in indoor, stereo image pairs, having large objects, apparently parallel to the camera image planes. Similarly in overlapping, constant luminous regions, all the block disparity values are likely to be zero.

The usual practice in coding the disparity value (i.e. in disparity compensated block based stereo image compression algorithms) is to use a fixed length code to indicate the disparity value of every block. However, under the above observations, this is not an efficient method of coding the disparity values.

Further analysis of a large number of disparity maps indicated that in most stereo image pairs vertically placed neighbouring blocks have a higher probability of having the same disparity value, than the horizontally placed neighbouring blocks. This is particularly true in cases where the image contains more vertically spread objects. In the proposed overhead bit coder, we make use of all the above observations to achieve a reduction in the total number of overhead bits. Under the scheme, if the block directly bellow an encoded block has the same disparity value as the encoded block, only one bit is used to indicate that its disparity is the same as in the previous block. If not, an additional amount of bits are used to indicate the new disparity value.

3.2.2 A Mathematical Approach

First, the conventional method of coding the disparity values for matching blocks is discussed.
3.2.2.1 Conventional Scheme

Assume that the search window size is \( N + 7 \) pixels wide, where, \( N \) is an integer. Thus, the disparity values (0 to \( N-1 \)) can be coded with the use of \( \log_2 N \) bits. Since the disparities are unidirectional, it is not necessary to have a sign bit. In the experiments carried out, the disparity values ranged from 0 to 63. Thus, 6 bits are necessary to represent these values. Let \( m \) be the total number of blocks to be encoded; then, each stereo image pair would need \( m \times \log_2 N \) overhead bits to code all the disparity values.

3.2.2.2 Proposed Scheme

The proposed method is explained as follows. If two adjoining blocks of the same column have the same disparity value, we send only a single bit to indicate that the second block has the same disparity value as the first. If it is not, we send, \( 1 + \log_2 N \), bits to indicate disparity; one bit is to say that it is different and the remaining \( \log_2 N \) bits to indicate the disparity value. Figure 3.3.1 shows the basic logic of the proposed overhead bit coder.

![Flowchart of the Novel Disparity Field Coder](image)

**Figure 3.3.1 Flowchart of the Novel Disparity Field Coder**
Thus, if a \( p \) amount of blocks have equal disparity values to the blocks immediately on top of them (in a column) in an image with a total of \( m \) blocks, \( p + [(m - p) \times (1 + \log_2 N)] \) overhead bits would be needed under the novel scheme. If this quantity is less than \( m \times \log_2 N \), which is the number needed if the conventional overhead bit coder was used, the use of the novel coder is justifiable. In order to arrive at a mathematical relationship between \( m, N \) and \( p \) we do the following simplifications.

\[
p + [(m - p) \times (1 + \log_2 N)] < m \times \log_2 N
\]

\[
\Rightarrow m - p \times \log_2 N < 0
\]

\[
\Rightarrow p/m > 1/\log_2 N
\]

In the experiments performed, \( N=64 \). Thus, \( p/m=1/6 \ (= 16.67\%) \), i.e., if more than 16.67% of the blocks have the same disparity value as the block above the proposed coder will be successful in achieving its purpose. Our experiments have shown that about 50% of the disparity values satisfied this condition in most of the images. Thus, a reduction in the search window size, even to an extent of 16 pixels wide (e.g. for low disparity stereo pairs), still justifies the use of the proposed coder.

### 3.2.3 Experimental Results & Analysis

To test the proposed overhead bit coder a set of five stereo image pairs derived from the web site, http://vision.stanford.edu/~birch/research/prp/ were used. All image pairs were of size \( 320 \times 320 \) pixels and were processed with a block size of \( 8 \times 8 \).

Table 3.2.1 below, shows the results obtained for the overall compression ratio of the DCTDP coding algorithm [17] using the conventional method of disparity field coding and the proposed method. It should be noted that the ECR figures presented here only illustrate the extra data compression for encoding all the right frames. In these experiments, it has been assumed that the disparity values of stereo image pairs will in general vary in the range \( 0-63 \) pixels [from \( 0 \) to \( N-1 \)]. This range was found to be adequate when the disparity values of a large set of stereo image pairs were investigated.
In accordance with the above assumption, we have taken six bits as necessary, to encode each disparity value, under the conventional scheme.

Table 3.2.1 Experimental Results – Novel Disparity Field Coder

<table>
<thead>
<tr>
<th>Stereo Image Pair</th>
<th>ECR % With Conventional Overhead Bit Coder</th>
<th>ECR % With Proposed Overhead Bit Coder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cans (a1 &amp; a2)*</td>
<td>25</td>
<td>33</td>
</tr>
<tr>
<td>Lamp (b1 &amp; b2)</td>
<td>18</td>
<td>26</td>
</tr>
<tr>
<td>Lab (c1 &amp; c2)</td>
<td>35</td>
<td>42</td>
</tr>
<tr>
<td>Packs (f1 &amp; f2)</td>
<td>38</td>
<td>44</td>
</tr>
<tr>
<td>Slanted</td>
<td>27</td>
<td>34</td>
</tr>
</tbody>
</table>

* The names within parenthesis are the original names given to the stereo pair in the web page

The results indicate that the proposed overhead bit coder gives an improved performance over the conventional, fixed length overhead bit coder. An average 7%, ECR increment is shown.

Table 3.2.2 Overhead Bit Compression Ratios

<table>
<thead>
<tr>
<th>N</th>
<th>Stereo Image Pair – OBCR %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cans</td>
</tr>
<tr>
<td>8</td>
<td>13.48</td>
</tr>
<tr>
<td>16</td>
<td>20.75</td>
</tr>
<tr>
<td>32</td>
<td>24.94</td>
</tr>
<tr>
<td>64</td>
<td>28.02</td>
</tr>
</tbody>
</table>

Table 3.2.2, shows the percentage compression achievable in the amount of bits necessary to code the disparity field, for each set of stereo images pair, under different search window sizes.

The Overhead Bit Compression Ratio (OBCR) is defined as a percentage as follows:

\[
OBCR = \frac{OB_{\text{Conv}} - OB_{\text{Prop}}}{OB_{\text{Conv}}} \times 100\%
\]
where, $OB_{\text{con}}$ and $OB_{\text{prop}}$ are the amount of overhead bits necessary if the conventional method of coding was used and the amount necessary for the proposed method, respectively. Mathematically this compression ratio can be proved to be equal to,

$$\frac{p}{m} = \frac{1}{\log_2 N}.$$ 

The overhead bit compression ratio curves plotted against the search window size $[N-1, 0]$ (horizontal search), $0$ (vertical search) for the five stereo image pairs are shown in Figure 3.2.2. Note that the curves indicate that the ratio $\frac{p}{m}$ remains constant.

![Figure 3.2.2 Performance Curves](image)

However, extensive investigations have shown that when the search window size is increased, $p$ decreases. This is due to the fact that a higher search range would decrease the probability of two adjacent blocks having the same disparity. This argument is justifiable, as a better match in terms of $MSE$ (although it may not be the proper block) would be more likely to be found within a larger search area. However, this new best match may not correspond to the proper location in terms of binocular parallax. This phenomenon is known as spurious block matching in literature.
The following disparity maps for the stereo image pair 'cans' illustrate this observation. The disparity map is formed on an 8×8 block basis with the disparity values mapped into the range [0-255]. Note that although most of the disparity values have remained unchanged, (when the search window size is increased) a small amount of blocks find better matches elsewhere. Thus, the value of ‘p’ above will decrease. This also has an effect on the performance curves illustrated in Figure 3.2.2, although they may be negligible.

![Disparity Maps for Stereo Image Pair 'Cans']
An interesting observation is that in contrarily to what would be expected after a visual inspection of the original image pair, the disparity field does not show a smooth variation in the background. However, further investigations showed this was due to the fact that the simulations were designed to pick up ‘the best match’ with an accuracy of five decimal points. In other words, it would have been possible to obtain a smoother disparity field in these areas, if a certain degree of flexibility was allowed in the selection of the best match. This would not incur a significant reduction in the overall compression efficiency, as the error blocks (in the pixel domain) formed would still be very low in magnitude, and would result in the same transformed and quantized block. This important observation has been used in the design of the object based stereo image compression algorithm proposed in Chapter 4.

3.2.4 Conclusions

An overhead bit coder for encoding the disparity values in disparity compensated block based stereo image compression algorithms has been proposed. The novel overhead bit coder makes use of the high probability of two adjacent blocks having equal disparity values, to achieve extra data compression as compared to using a conventional, fixed length overhead bit coder. Experimental results have proved that an average 7% increment of the compression performances can be achieved for the test image set under consideration.

Some important observations have been made as a result of this study. These have been used in the design of an object driven, block based stereo image compression algorithm (Chapter 4) which removes the inherent disadvantages of exclusively block based compression algorithms.

3.2.5 Improvements, Applications and Further Research

In the above method the strategy used was to code a disparity value using a fixed length code, if that value was different from the disparity value of the block above it. An alternative approach to this would have been to code the difference of the two disparity values (differential coding) under such circumstances. As the difference would likely be
less in magnitude than the disparity value itself, if a variable length code was used to code these differential disparity values, we would expect to get further improvements in compression performance. However, experiments with such a scheme indicated that the additional gains achievable are marginal and is not worthy of the extra complexity introduced. Two major reasons for this behaviour were observed. The first is that the introduction of a variable length code makes it a requirement that overhead bits be transmitted as end of disparity value codes. As the disparity values typically vary between 0-63 pixels, if a fixed length code was used only up to 6 bits would have been used for coding the disparity value. If a variable length code was used, the length of the end of disparity value code is not significantly lower than 6 or the average code word length, thus making the overall gain achievable with this strategy marginal. The second reason is that due to spurious matching in some stereo image pairs the difference between adjacent unequal disparity values may require more bits for it’s binary coding. Spurious block matching in disparity compensated stereo image compression may result due to noise, luminance disparity resulting from reflections and left/right camera differences. This is also the basic factor that would limit the performance of the proposed algorithm in Section 3.2.

In MPEG video coding standards [145,146] differential coding has been used in coding the motion vector field. However, in the above paragraph it was explained that such a coding scheme would not be of significant importance in coding disparity fields. The reason lies in the fact that motion vector fields are far smoother than the disparity vector fields. The complex nature of some stereo image scenes would give rise to dense disparity maps, making differential coding of disparity fields less effective.

One important observation of this research was to experimentally conclude the behaviour of the disparity field in order to suggest more efficient ways to compress stereo image pairs. It was concluded that methods that would segment a given image pair into areas, regions or objects of constant disparity, would vastly improve the compressibility of the stereo image pair. Block based approaches such as the DCTDP coding scheme [17] fail to address this issue.
3.3 Pioneering Block-Based Predictive Coding

This chapter introduces the reader to two novel predictive coding algorithms for stereo image compression. In the proposed schemes, predictive coding is applied to one of the two frames associated with each stereo image pair on top of DCT/Quantization based compression technology. These novel coding techniques eliminate the necessity of using overhead bits in reconstructing the predicted image.

3.3.1 Introduction

As described in Chapter 1 and Chapter 2, the existing stereo image compression schemes, either take advantage of the psychovisual aspects of the human vision system [31,32,51,65], or extend the de-correlation process of monocular compression schemes to spatial, temporal, as well as left/right image stream correlation domains. The proposed predictive coding algorithms fall into the second category stated above. They make use of the high correlation between the image data itself (intra-frame, spatial) and the high correlation between the left and right frames (inter-frame) to achieve data compression.

Existing technology in predictive coding can be broadly classified into two groups, namely, linear predictive coding [150,153] and non-linear predictive coding [154-157]. Linear predictive coding involves establishing a predictive pattern weighted by a number of coefficients that are then used to produce predictive values. JPEG loss-less image compression mode is one typical example of this type. Other examples are those techniques developed in telecommunications, such as Differential Pulse Code Modulation (DPCM) and Adaptive Differential Pulse Code Modulation (ADPCM). Non-linear predictive coding varies from one algorithm to the other. Representative examples include neural network based predictive coding [154,155] and Vector Quantization based predictive coding [157]. All these techniques are developed based on the fact that images are digitised from analogue signals and neighbouring pixel data are correlated. However, this predictive coding technology is not suitable to exploit the extra correlation present in stereo image pairs and achieve the best possible data compression. The reason being that they only make use of spatial and temporal redundancies but not the third dimension of redundancy present in stereo image pairs. The proposed algorithms use this additional dimension of redundancy to achieve extra data compression.
Note that the JPEG still picture compression standard has been used as the basic compression engine for the algorithms proposed. Further refinements to the algorithms are possible by introducing other schemes (e.g. wavelets) to replace JPEG. Results reveal that up to 28% extra compression (ECR %) is achieved on top of JPEG compression for the frame predicted, i.e., for the right frame.

The rest of this chapter is arranged as follows. Section 3.3.2 contains a comprehensive discussion into the novel predictive coding algorithms proposed for stereo image compression. Section 3.3.3 analyses the results of the simulations carried out. Section 3.3.4 proposes a modification to the encoder of the basic algorithm to guarantee the reconstruction of the predicted image. Section 3.3.5 provides a conclusion.

### 3.3.2 Algorithm Design

The first algorithm uses a single 8×8 image block as the guiding block (*pioneering block*) to search the left frame for a best match and the second uses the average of two such pioneering blocks for the same purpose. Both algorithms are based on the same principle as discussed later.

For the convenience of presentation, two *domains* are defined. The first is *pixel domain*, which means that image frames are not transformed or compressed, and are in their original data format. The second is *frequency domain*, which means that the images are transformed into DCT coefficients and scalar quantized as proposed by JPEG.

The overall encoder and decoder block diagrams are illustrated in Figure 3.3.1 and Figure 3.3.2 respectively. These are for the single pioneering block predictive codec. Slight modifications to these schematic diagrams would be necessary for the codec representing the algorithm that makes use of two pioneering blocks.
We define the following terms and functions before describing the details of the compression schemes.
\([L] = \{x_{ij}\}\), An image block from the original left frame.

\([R] = \{y_{ij}\}\), An image block from the original right frame.

\([L]\) = \{x'_{ij}\}\), An image block from the reconstructed left frame.

where, \(x_{ij}, y_{ij}\) (1 \(\leq i, j \leq 8\)) denotes pixels from the left frame and the right frame respectively.

\(d\) - a function which finds the squared distance between two image blocks.

\(f\) - a function which converts a given block from its pixel domain to its quantized frequency domain.

\(f_1^{-1}\) - a function which converts a given block from its quantized frequency domain to its pixel domain.

Note that \(f_1^{-1} \neq f^{-1}\) because of the rounding-off process associated with scalar quantization.

Thus, we have the following relationships between the above terms,

\[
[L] = f_1^{-1}(f([L])) \quad (3.3.1)
\]

\[
d([R], [L]) = \sum_{i=1}^{8} \sum_{j=1}^{8} (y_{ij} - x'_{ij})^2 \quad (3.3.2)
\]

In addition, we define a co-ordinate system for image blocks as shown in Figure 3.3.3.
3.3.2.1 Algorithm 1 – Single Pioneering Block Coder

This algorithm, uses a single pioneering block in the prediction of the block to be encoded both at the encoder and decoder ends. The block selected as the pioneering block is the one that precedes the block to be encoded. For clarity of presentation the algorithm is explained in several steps.

(a) Selection of search window

In Section 2.4.4.1 it was shown that stereo image pairs obtained using parallel axis camera geometry have two special properties associated with stereo disparity. The first property being that, there is no vertical disparity between the matching points in the frames. The second property is, that all points in the left frame are displaced only to the right of the corresponding object locations in the right frame, i.e., the horizontal disparity present in such a stereo pair is unidirectional. Thus, if \([R]_{m,n}\) (\(m\)-column number of the block array, \(n\)-row number of the block array) is the block to be encoded in the right frame, the best possible match, \([L]_{\text{match}}\), in the left frame satisfies the following condition:

\[
[L]_{\text{match}} = [L]_{k,n} \quad k \geq m
\]  

(3.3.3)
(b) Selection of pioneering block

To encode $[R]_{m,n}$ in the right frame, the block immediately preceding it, $[R]_{m-1,n}$, is used as a pioneering block to search for the best match in the left frame. This selection is justified due to the presence of high intra-frame correlation, i.e. we can expect the preceding block to be highly correlated to the block to be encoded.

(c) Disparity compensation

For accuracy of prediction and increased speed, we limit the search area to a window in the left frame. Due to the unidirectionality of the disparity, this window only needs to spread to the right of the block in the left frame, which has the same location co-ordinates as the selected pioneering block. A search window of 64 pixels wide was found to be sufficient for most of the test stereo image pairs. The disparity compensation is done by comparing the pioneering block with all the blocks available within the search window, under a least mean scared matching criterion. The presence of high inter-frame redundancy between the stereo pair would result in the selection of an accurately matching block to the pioneering block from within this search window.

(d) Prediction

After the best match is found from the reference frame the block to the right of it is chosen as the predictor to the block to be encoded. This block then attends the linear predictive coding process described in Section 3.1 to produce errors. For clarity of presentation this is not indicated in Figure 3.3.1 in detail. Note that as the linear prediction is done in the frequency domain the best match and the block to be encoded has to be transformed into the frequency domain as an initial step. Each time a block is encoded, it becomes the preceding block and attends the same search operation to find its best match and provide guidance for encoding the next block.

Both the left frame and the errors are compressed by JPEG, i.e., DCT and scale quantization, followed by entropy coding [141]. The decoder also works in a similar manner. Since at the decoder, the reconstructed left frame and the preceding block are readily available, overhead bits that provide information about the location of the best
match need not be encoded. In other words the decoder can easily locate the best match from all the data decoded and available to it.

(e) A mathematical representation

A vigorous mathematical representation of the proposed predictive coder is as follows.

Assume that a particular block, say \([R]_{m,n}\), \((1 \leq m \leq p \text{ and } 1 \leq n \leq q}\) where \(p\) and \(q\) are the number of blocks in a row and number of blocks in a column, respectively) needs to be encoded. First choose the block immediately preceding it, i.e., \([R]_{m-1,n}\), as the pioneering block to search the left frame for a best match.

Note that for those blocks located in the first column of the right frame (i.e. when \(m = 1\)) however, special care is required where the block right on top of the block to be encoded (i.e. block \([R]_{m,n-1}\)) needs to be selected as the pioneering block. For blocks near the right end of the frame, the search area is chosen to be of the same width, but a window that spreads from the right corner of the frame to the left. This is done only to reconstruct the right edge of the right frame.

For the specific searching operation, the selected pioneering block from the right image, \([R]_{m-1,n}\), is matched against every possible block within the search area in the left frame. In other words, new blocks for matching are selected by shifting the previously matched block in the left frame by one pixel horizontally to the left each time. A squared error based matching criterion is used.

It has to be noted that there is a possibility of selecting two or more blocks with identical minimum distance value to the pioneering block. In such a case, we assume with reasonable accuracy that the 'correct match' should most likely be the block which is located closest to the corresponding location of the pioneering block in the right frame. Thus, we choose the block closest to the left end of the search window since this block corresponds to the location of the pioneering block in the right frame. Thus, every time a block is selected for matching, its displacement from the left end of the search window
and its MSE compared with the pioneering block is recorded. The best match, \( [L_{\text{match}}] \), is selected to satisfy,

\[
d([R,L_{\text{match}}]) = \min_{[L] \in W} d([R,L])
\]

(3.3.4)

where, \( d([R,L]) \) is given by equation (3.3.2) and \( 'W' \) denotes the search window. Let this best match be \( [L]_{a-1,n} \) in accordance with the block co-ordinate system for the left frame defined in Figure 3.3.3. \( [L]_{a,n} \) is then selected as the predictor for the block \( [R]_{a,n} \).

In the case that \( [R]_{a,n} \) is located in the first column of the right frame, the predictor will be the block immediately below the best match. The complete search procedure is illustrated in Figure 3.3.4.

\[
E_t = f([R]) - \{ A \otimes f([L]) + B \} 
\]

(3.3.5)
where, $A$ and $B$ are the coefficients of the linear predictor used in improving the performance of the prediction stage, as described in Section 3.1. Note that the block location co-ordinates have been dropped in $R$ and $L$ for the clarity of presentation.

These errors are then entropy coded and transmitted. The left frame image blocks are transformed into their frequency domain (i.e. $f(L)$) is found for each image block) and then transmitted after entropy coding. As overhead bits are not required for the decoding process, additional hardware is not necessary at the decoder to differentiate between overhead and error codes. This is one of the practical advantages of the proposed algorithms.

The decoding process can be described in a similar manner. At the decoder (see Figure 3.3.2), the left frame is reconstructed by using Equation 3.3.1. The right frame is reconstructed as follows: Assume that the first block at the top left corner of the right image is initially known. This block is taken by the decoder as the pioneering block to decode the block immediately to the right of it by following the same searching and matching operations as at the encoder end. After the decoder finds the best match for the pioneering block from within the search window of the already reconstructed left frame, it selects the block right after it as the predictor. Then it adds the received prediction errors of the corresponding location to this predicted block (after linear compensation) to form the decoded block. Note that this is done in the frequency domain. This procedure is continued for all the blocks in the right frame until it is fully recovered. Each time a block is decoded, it becomes the pioneering block for decoding the next block. Decoding blocks in the first column of blocks is explained in a similar manner to that of the encoding process.

Experimental results indicated that there are a few instances where finding the best match for the pioneering block may not lead to finding the best possible match for the block to be encoded. Further experiments highlighted two possible reasons for this behaviour. It was observed that in areas where adjacent blocks are less correlated or in partially occluded areas one pioneering block does not provide sufficient information to find the ‘correct’ best match. This was also found to be true at instances where more than one best match is found for a given pioneering block. The simple way of choosing the best
matching block (out of the equally matching ones), closest to the left end of the search window is sometimes not correct although it simplifies the searching procedure.

Thus a second algorithm, which uses two pioneering blocks, was proposed and tested.

### 3.3.2.2 Algorithm 2 – Two Pioneering Block Coder

This algorithm is also based on the same basic principle that stereo image pairs contain both high inter-frame and intra-frame redundancy. The only difference is the fact that two pioneering blocks are now selected for searching and matching processes, as illustrated in Figure 3.3.5, instead of the single block considered in Algorithm 1.

![Figure 3.3.5 Search Procedure - Algorithm 2](image)

For the block to be encoded, denoted as \([R_{m,n}]\), we select the block preceding it, \([R_{m-1,n}]\), and directly above it, \([R_{m,n-1}]\), as the two pioneering blocks. This choice is justified in the sense that these blocks will be highly correlated to the block to be encoded for being its immediate neighbours. Once the two blocks are selected their pixel domain average is taken. This average, denoted as \([R_{\text{avg}}]\), is then tested against the pixel domain average, say \([L_{\text{avg}}]\), of similar block pairs in the left frame for the best matching average, \([L_{\text{match}}]\). \([L_{\text{match}}]\) would satisfy the following equation:
Thus, if blocks $[L_{\text{match}}]_{a-1,n}$ and $[L_{\text{match}}]_{a,n-1}$ were the blocks which resulted in the best matching average, $[L_{\text{match}}]$, we take $[L]_{a,n}$ as the predictor for $[R]_{a,n}$. In the above matching process, the search for the best match for the preceding pioneering block, inside the left frame is again limited to a similarly shaped search window as in Algorithm 1. The search for the best match of the other (top) pioneering block is done on a window displaced one block location to the top and one block location to the right relative to this search window. As the image blocks would be reconstructed in a raster scanning order both pioneering blocks will be available at the decoding end to reconstruct the right frame on a block-by-block basis.

However, it has to be noted that Algorithm 2 will not work for the first row of those blocks in the right frame, since no block can be found as the top pioneering block, $[L]_{a,n-1}$. In this case, we use algorithm 1 to predict this row.

### 3.3.3 Experimental Results

Both algorithms were tested against stereo image pairs obtained using parallel axis camera geometry. The majority of these stereo image pairs were obtained from WWW sites [148,149]. All test images were of size: 320 x 320 pixels. The simulations were carried out using purpose built MATLAB routines. Some sample results are presented in Table 3.3.1 & Table 3.3.2. The ECR % tabulated is defined as in Section 2.6.1. A search window size of [0-63] was assumed.
Table 3.3.1 Compression Performance of Algorithm 2

<table>
<thead>
<tr>
<th>Image Pair</th>
<th>DCTDP Algorithm ECR %</th>
<th>Proposed Algorithm 2 ECR %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Castle</td>
<td>34</td>
<td>38</td>
</tr>
<tr>
<td>Corrugated Mat</td>
<td>36</td>
<td>40</td>
</tr>
<tr>
<td>Columns</td>
<td>36</td>
<td>36</td>
</tr>
<tr>
<td>Cans</td>
<td>25</td>
<td>50</td>
</tr>
<tr>
<td>Study Lamp</td>
<td>18</td>
<td>46</td>
</tr>
<tr>
<td>Lab</td>
<td>35</td>
<td>48</td>
</tr>
<tr>
<td>Packs</td>
<td>38</td>
<td>43</td>
</tr>
<tr>
<td>Slanted</td>
<td>27</td>
<td>47</td>
</tr>
</tbody>
</table>

Table 3.3.2 Comparing Performance of Algorithm 2 against Algorithm 1

<table>
<thead>
<tr>
<th>Image Pair</th>
<th>JPEG Bit No</th>
<th>Algorithm 1 Bit No</th>
<th>Algorithm 1 ECR %</th>
<th>Algorithm 2 Bit No</th>
<th>Algorithm 2 ECR %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Castle</td>
<td>97,792</td>
<td>062,349</td>
<td>33</td>
<td>057,525</td>
<td>38</td>
</tr>
<tr>
<td>Cor. Mat.</td>
<td>90,018</td>
<td>059,447</td>
<td>34</td>
<td>053,591</td>
<td>40</td>
</tr>
<tr>
<td>Columns</td>
<td>87,839</td>
<td>060,872</td>
<td>30</td>
<td>057,447</td>
<td>36</td>
</tr>
<tr>
<td>Land</td>
<td>172,272</td>
<td>115,772</td>
<td>34</td>
<td>107,155</td>
<td>37</td>
</tr>
<tr>
<td>Earth</td>
<td>154,221</td>
<td>104,870</td>
<td>32</td>
<td>099,194</td>
<td>36</td>
</tr>
</tbody>
</table>

Results in Table 3.3.1 indicate the superiority of the proposed algorithms as compared to the DCTDP coding scheme. Note that the improvements obtained for the compression ratios are dependent on several factors. For the proposed techniques the gains are obtained due to the fact that they do not require the transmission of overhead bits to indicate the disparity values. However, the use of the pioneering blocks in the disparity compensation, may result in a less accurate block being predicted for the block to be encoded, as compared to what would be predicted under the DCTDP coding scheme. Our experiments indicated that this happens only in areas where intra-frame redundancy is very low, where the pioneering blocks may not be highly correlated to the block to be encoded. Table 3.3.1 also indicates that certain image pairs (Cans, Study Lamp &
Slanted) perform better than the others in terms of percentage improvement of compression ratios. These images contain relatively large areas with low levels of depth information and small areas with large amount of depth information, making the percentage of overhead bits higher compared to the error bits needed under the DCTDP coding scheme. Note that a fixed length code has to be allocated for coding the disparity values and the length of this code would depend on the maximum binocular parallax present in the scene.

Results in Table 3.3.2 indicate that Algorithm 2, which uses two pioneering blocks, outperforms Algorithm 1. It was observed that the introduction of a second pioneering block in Algorithm 2, increased the chances of a better matching predictor to be found for the block to be encoded, using pioneering blocks as the guides. This is justifiable due to the fact that the two pioneering blocks that have been selected would define the block to be encoded and also it’s best match more accurately, as compared to the one pioneering block scheme. This directly results in an improvement of the compression performance of Algorithm 2 as compared to Algorithm 1. This is indicated by the results in Table 3.3.2.

Figure 3.3.6 below, shows the results for the test image set ‘Castle’ with Algorithm 2 used for the compression of the right frame. The left frame, which acts as the reference, has been compressed up to 80% using JPEG. This ratio is defined as the total number of output bits from the JPEG encoder divided by the number of input bits for the original image, if 8 bits were used to code the intensity value of each luminance pixel. As the predictive coding is lossless the output image quality is only subject to the JPEG image quality. Thus, under circumstances of reasonable JPEG quality accurate reconstruction of the right image is guaranteed.
Further experiments revealed that the proposed algorithms are sensitive to the JPEG compression quality since the encoder works on the original left image, yet the decoder works on the JPEG reconstructed left image. In other words, the lossy effect of JPEG compression which is used to compress the left image and the prediction errors, may introduce an image quality reduction because of the possibility that the decoder could possibly choose a different best match as compared to that of the encoder. This in turn would reconstruct a wrong block at the decoder end. This image quality reduction is
subject to the level of quantization used or the extent of information loss in JPEG. It was observed, however, that if the quantization step sizes were maintained at a reasonable level the right image was accurately reconstructed with a quality level equivalent to that of the reconstructed left frame. This was illustrated in Figure 3.3.6.

3.3.4 Improved Encoder

In circumstances where JPEG compression quality is very low, the encoder can be modified as shown in Figure 3.3.7 to guarantee the accurate reconstruction of the right image. For the purpose of easy reference the decoder illustrated by Figure 3.3.2 has been reproduced in Figure 3.3.8. Note that two modifications have been made in the improved encoder of Figure 3.3.7 in comparison with the encoder illustrated in Figure 3.3.1.

The first is to reconstruct the left image using a JPEG decoder at the encoder end and to use this reconstructed image in the encoding of the right frame blocks. This implies that the search for the best matching block at the encoder and decoder end is carried out on identical reference images. This is one of the two essential criteria in guaranteeing the reconstruction of the right image under any JPEG compression quality. The second criterion is to reconstruct each encoded block, using the prediction error and the predicted block, at the encoder end. This reconstructed block would act as the pioneering block for the next block to be encoded. Thus, while the right image is encoded block by block in a raster scanning order, it is also reconstructed at the encoder end. This partially reconstructed (up to the previously encoded right image block) right image is used to find the pioneering block/s, once a decision is made to encode a particular block. However, the original pixel values of the current block being encoded are used in the calculation of prediction error. In the algorithm where two pioneering blocks are used, the reconstructed image blocks would need to be stored at least up to the block above the location of the current block to be encoded. In other words if the image contains $N$ image blocks in a row, an $N$-block store would be necessary. Note that under such circumstances Figures, 3.3.7 and 3.3.8, will need modifications in their structure.

If the above criteria are observed, the two images involved in all the operations of the block-based search and prediction will be the same for both encoder and decoder. Hence, accurate reconstruction of right image is guaranteed.
Figure 3.3.7 Modified Encoder

Figure 3.3.8 The Decoder
The placement of the quantization/de-quantization, DCT/Inverse-DCT, stages relative to
the prediction stage in Figure 3.3.7 and Figure 3.3.8, is equally important in the
guaranteed reconstruction of the right frame. This is explained (for Algorithm-I) using
the following mathematical explanation.

Let,
\[ q \] = a function representing scalar quantization
\[ q^{-1} \] = a function representing inverse-scalar quantization
\[ dct \] = Discrete Cosine Transform
\[ idct \] = Inverse DCT
\[ R \] = Right image block to beEncoded (note: in original pixel domain)
\[ R' \] = Right image block, reconstructed pixel domain.
\[ L' \] = Best matching block from reconstructed left image in pixel domain.
\[ e \] = Error block in frequency domain.
\[ A, B \] = Linear predictor coefficients.

Thus, at the Encoder end (Figure 3.3.7) the output of the prediction stage (at callout No.1)
would be,

\[ e = q(dct(R)) - \{A \times \{q(dct(L'))\} + B\} \]

The output at 'callout No. 2', is
\[ R' = idct(q^{-1} \{e + \{A \times \{q(dct(L'))\} + B\}\} \]
i.e. \[ R' = idct(q^{-1}\{q(dct(R))\}) \]

\[ R' \], now acts as the pioneering block for the next block to be encoded. As the Encoder
and Decoder does the search in the same reconstructed left frame (JPEG coded &
decoded), as long as the same \[ R' \] block is produced at the Decoder end to be used as a
pioneering block, the reconstruction of the right frame would be guaranteed.

At the Decoder end, assume that the previous block has been decoded accurately. This
means that when it is used as the pioneering block to decode the current block, it will find
the same block, \( L' \) from the reconstructed left frame, as the best predictor. Thus, at the decoder end, the input at ‘callout No. 3’ (i.e. the error) would be,

\[
e = q(dct(R)) - \{A \times \{q(dct(L'))\} + B\}
\]

and the output at ‘callout No. 4’ would be,

\[
e + \{A \times \{q(dct(L'))\} + B\} = q(dct(R))
\]

Thus, the output at ‘callout No. 5’ would be,

\[
= idct(q^{-1}\{q(dct(R))\}) = R'
\]

Thus, in the decoding of the next block, \( R' \) would be used as the pioneering block. As this is identical to the block that would be used as the pioneering block to search for the best match, in the next block to be encoded, by mathematical induction it is proved that this process would guarantee the reconstruction of the right frame. However, if the quantization/de-quantization, DCT/inverse-DCT stages were erroneously placed with respect to the prediction stage, this would have not been possible.

The above mathematical approach also proves an important property about the quality of the reconstructed right image. That is, the reconstructed image is identical to what would have been obtained if the right frame was directly transmitted using JPEG. The quality would only depend on the scalar quantizer used. However, in DCTDP coding algorithm, the reconstructed right image is not exactly the same as this. This is due to the fact that disparity compensation at the Encoder end is done on the original left frame and the disparity vector based prediction at the decoder end is done on the reconstructed left frame. The end result is a reconstructed right frame whose quality is a function of the scalar quantizer, linear predictor coefficients \( A \) and \( B \), and the image quality of the reconstructed left frame.

However it is not possible to clearly state whether the reconstructed right image quality under the DCTDP scheme is better or worse as compared to the proposed technique with
the modified Encoder. Experimental results indicated mixed results, with less than 0.5 dB PSNR differences between the two.

The compression performance, however, will be slightly reduced due to the fact that the blocking effect inherent in JPEG compression would reduce the intra-frame correlation in both left and right frames. Experimental results indicated that this reduction is in the range of 0.5-2.0 % for the results shown in Table 3.3.1.

3.3.5 Conclusions

In this section, two novel predictive coding algorithms for stereo image compression were introduced. These techniques eliminated the necessity of using overhead bits in reconstructing the predicted image in block based stereo image compression, thus obtaining compression gains of up to 28% compared to the widely used benchmark algorithm, the disparity compensated transform domain predictive coding scheme. Conditions that lead to image quality loss under these schemes were identified and necessary solutions were designed, tested and implemented in section 3.3.4. Some special properties of the proposed modification of the algorithms have been theoretically and experimentally proved. The algorithms have been later used in conjunction with the proposed object based stereo image compression scheme (Chapter 4) for achieving best image quality / compression gain, performances.

3.3.6 Improvements, Applications and Further Research

The complexity and the computational cost of the simple (without encoder modification), single pioneering block based predictive coding encoder is equivalent to that of the benchmark DCTDP coding scheme. In fact as the disparity field need not be coded the encoding algorithm is less computationally costly than the DCTDP coding scheme. However, as the pioneering block scheme requires disparity compensation to be done at the decoder end as well, the decoding process of this scheme is more complex as compared to the straightforward (disparity field based) decoding process of the DCTDP scheme. But, here again, decoding of the disparity values would not be necessary. However, the overall effect would be of a slightly slower decoding process for the pioneering block scheme. The speed of a stereo image codec in real time applications
such as video conferencing, real time imaging, etc., would be limited by the slowest process (whether it is encoding, decoding or transmission). As the above pioneering block encoder is faster than the encoder of the DCTDP scheme, and the encoder and the decoder of the pioneering block scheme are identical, the scheme comes handy in real time applications. The main advantage being in the extra compression obtainable.

The computational cost of the two pioneering block based predictive encoder is higher than the single pioneering block encoder. This is due to the fact that the matching is done by comparing the average of two pioneering blocks to the average of two candidate matches in the reference frame, search window area. Thus, for every block to be encoded an additional \((X+1)\), 8\(\times\)8 matrix summation operations, would be performed. Here \(X\) is the amount of candidate matches against which the average of the two pioneering blocks is compared. However, the speed of this implementation could be vastly improved by reusing the major part of 'element to element summation' of the two candidate blocks. As the search window area is only horizontally spread, between two consecutive candidate block averaging steps, only one extra column of pixels need to be averaged as columns. No. 2-8 of the previous pair blocks would now form column numbers 1-7 of the new pair of blocks with only the 8\(^{th}\) column pixel averages being new. Apart from this, as adjacent blocks to be encoded have overlapping search window areas in the reference frame, the above averaging steps could be further reduced. Thus, the computational complexity of the basic two pioneering block algorithm, when used in real time applications would only be slightly higher than that of the DCTDP coding algorithm. However, for off line imaging applications where the speed of the decoding process becomes the limiting factor, straightforward implementations of the pioneering block schemes are slower in speed.

An alternative way to reduce the computation cost of the two pioneering block scheme is to first use one pioneering block only (the block in front of the block to be encoded) to find a pre-specified (say best four matches or matches that fall below a certain MSE threshold) amount of best matching candidate blocks from the reference frame. After this, the effect of the corresponding second pioneering block is considered only for those selected first pioneering blocks. This would prevent unnecessary calculations for matches that would be far from accurate.
Further experiments carried out by M.Bax and A.Vitus, of Stanford University, USA has resulted in the detection of an important feature of the above schemes, which is best illustrated by the rate distortion curve in Appendix 1. Bax and Vitus have used a coding strategy in which the prediction errors of blocks that find reasonable matches from the reference frame (below a threshold) are not transmitted. Under such a scheme it has been observed that this encoder can achieve a signal-to-noise ratio at the receiving end of greater than 15 dB without transmitting any bits for the prediction of the right image. This research work recommends the proposed coder as the best existing method for very low bit-rate applications. The algorithm has been compared with the most of the efficient existing stereo image pair compression schemes, namely: pixel-based disparity map encoding, block-based disparity map encoding (DCTDP coding scheme [17] used as the benchmark), and compression using sub-sampling and transform coding. [17,122].

Experimental results tabulated in Table 3.3.2 show that using two pioneering blocks result in a better compression performance as compared to using one pioneering block. Under the simple pioneering block algorithm, this also results on a more accurate right frame reconstruction. Under high levels of JPEG image quality loss in the reference, left frame, the single pioneering block scheme mainly fails near high luminance gradient edges. The reason for this is that the DCT/scalar quantization based JPEG compression in these areas cause higher degradation of image quality as compared to the smoothly textured areas. This causes the correlation between adjacent blocks in such areas in the reconstructed reference frame to reduce, as compared to the correlation between the blocks at the same location in the original reference frame. Extensive experiments indicated that even the simple pioneering block encoder with one pioneering block performed extremely well under high JPEG image quality loss of the reference frame, in smoothly textured areas. This observation was successfully used in using this algorithm in the internal area coding of certain objects in the object based stereo image compression algorithm proposed in Chapter 4.

The pioneering block algorithms are based on the fact that adjacent blocks within a given image are correlated. This is a reasonable assumption for most images under normal noise levels. However, the performance of the algorithm would reduce in the presence of extreme noise levels.
In section 3.3.4 an improvement to the basic pioneering block encoder that would guarantee the reconstruction of the predicted frame was proposed. It was proved theoretically that the reconstructed image is identical to what would have been obtained if the right frame was directly transmitted using JPEG. The quality was found to only depend on the scalar quantizer used. The compression performance of this modified codec was found to be 0.5-2.0% less as compared to the simple pioneering block codec as the matching between the left frame in the reconstructed domain and the right frame in the original would lead to slightly higher prediction errors. MPEG standards [145,146] use a similar strategy in predicting in the temporal domain. The best match for a block in the current frame is found from the reconstructed previous I frame or P frame. However, as the motion vectors are transmitted along with the prediction errors the aim of doing this is not, for guaranteeing the proper reconstruction of the current frame. Rather, the idea is to obtain a measure of the reconstruction error of the block at the encoder end so that different strategies could be applied for coding. Although the modified encoder is more computationally complex as compared to the simple pioneering block encoder, it guarantees the reconstruction of the right frame under any level of JPEG image quality loss in the left frame. This makes it suitable for very low bit rate applications where the left frame too would have to be compressed to excessive levels.

The pioneering block based predictive coding scheme can equally well be applied for video compression. In this case the left frame (reference frame) would be substituted by the previous frame in the spatial domain and the right frame (predicted frame) would be substituted by the current frame. Initial experiments carried out in replacing the motion vector based predictive coding scheme of MPEG-2 coding standard by the pioneering block based predictive coding algorithm has provided encouraging results in terms of coding efficiency. Some implementation issues that would speed up the new coding scheme, such as what was described above in re-using some pixel averages, are currently under investigation. Although the basic search procedure is more complex, the pioneering block scheme has the advantage that motion vectors need not be coded and transmitted. This saves computing time and resources and is expect to balance the computational complexity introduced due to the averaging of the candidate best match pair.

The pioneering block based predictive coding algorithms proposed in Section 3.3 are exclusively block-based algorithms and are thus useful in relation to the development of
MPEG-1, MPEG-2 based stereo sequence compression algorithms. Both these coding standards would vastly benefit by a stereo video-coding version that does not involve the transmission of additional overheads in the form of disparity information. Further research work is encouraged in this direction.

The use of the pioneering block algorithm with variable size [53,127] and deformed shape [127] block-based coding schemes that use quad-tree decomposition [13,14] is encouraged. These algorithms deal better with occlusion [6,48,75] and perspective distortion as compared to fixed sized block based stereo image coding algorithms. However, fixed sized block processing is popular in modern image [141-143] and video coding [145] standards due to the simplicity in design and implementation. One solution to improving block matching with fixed sized blocks is to reduce the block size. However, this means that more overhead bits need to be transmitted as disparity (in stereo) or motion (in video) vectors. The pioneering block schemes come handy under these circumstances.

It is also recommended to assess the performance of the pioneering block scheme when the DCT transform is replaced by wavelet transform. Better quality, reconstructed images are expected by this replacement, especially because the DCT transform based schemes suffer from blocking artefacts.
Chapter 4

An Object Based Algorithm for Stereo Image Pair Compression

4.1 Introduction

Fundamentally, there are two different ways to estimate the disparity field of a stereo image pair: the intensity based method and the object/feature based method. The first looks for correspondence between luminance values, and the second determines a set of objects/features in both images and seeks correspondence between the two sets.

The simplest intensity based method is block matching, in which the image frames are divided into sub-blocks of an appropriate size and are matched under a certain matching criterion for maximum correspondence. Several authors [8,15,17,31,125,163] have proposed block-based stereo image compression algorithms. The pioneering block based stereo image compression algorithm proposed in Chapter 3 falls into this category. These algorithms have the advantage of being simple and being easily adaptable to be used in association with standard image compression techniques (JPEGs and MPEGs) [141-145]. However, they suffer from inefficient exploitation of the inter-frame redundancy that is unique to stereo images in the sense that disparity is determined via fixed sized blocks rather than objects. Yet human visual perception of depth in stereo images is exactly interpreted in terms of objects rather than rigid blocks. Therefore, it is unlikely that block-based techniques could achieve any further improvement in comparison with the existing state-of-the-art developments. In addition, it also fails to address the compression strategy in an object-based perspective, which is the core in the development of modern and future interactive communication systems [146]. As a result, several authors have made some successful attempts [29,43,52,63,81,86,104,106,136] to deviate research on stereo image (still and sequence) compression from the more conventional block based approach to an object/feature-based approach.
In this chapter, a novel hybrid approach between block and object based techniques for the compression of stereo image pairs, is proposed. The basic idea behind the development of this algorithm is the efficient exploitation of redundancy in smoothly textured areas that are present in both frames, but are relatively displaced from each other due to binocular parallax. The algorithm has been designed based an object-layered platform, in a similar spirit to that of MPEG-4 [146]. Although the algorithm has been designed and tested for stereo image pairs, rather than for sequences, the ideas can be easily extended in the temporal dimension.

4.2 An Overview

Figure 4.1 shows the block diagram of the proposed stereo image encoder. The left frame, which is selected as the reference frame, is independently compressed using a JPEG encoder [141-143], as described in Section 2.3. It is then reconstructed locally at the encoding end in order to achieve a guaranteed correct decoding of all the right frames.
The proposed algorithm performs object extraction and matching between the reconstructed left frame and the original right frame to identify those objects or areas that match but are displaced by varying amounts due to the binocular parallax. The first step towards achieving this is to use a contour extraction process (Section 4.3.1) [161]. Depending on one or more pre-selected intensity thresholds, an edge detection step [159-161,165] followed by a contour tracing strategy [161] is adapted on both frames to identify objects or areas that have similar texture. The resulting contours are matched based on their shape information to find those objects/areas, which are similar in shape (Section 4.3.2).

The matching object pairs are then included in their tightest possible bounding rectangles (Section 4.3.3). These rectangles are suitably extended in order to be divided into 8x8 sub-blocks. The right bounding rectangle is then adjusted to the size of the left object bounding rectangle which acts as the reference in predicting the right object. The resulting rectangles are identified as object planes. Note that the centres of areas (centroids) of matching objects are calculated and used when the bounding rectangles are defined and aligned. The next step is to classify the constituent 8x8 sub-blocks on both object planes into three types namely, interior blocks, boundary blocks and exterior blocks (Section 4.3.4). The boundary blocks are those that have at least one of the contour points within them and thus are easily identified. A parity-based contour filling algorithm [158,159] is used to identify the interior and exterior blocks (Section 4.3.4). At the end of this step, the shapes of the left and right objects are also identified as binary object planes. These planes are used as an aid in the coding process.

As a result, the identified objects are further categorised into three groups: (i) matching object pairs that are enclosed by closed contours (Section 4.3.2.1); (ii) matching object pairs enclosed by contours that terminate at the image frame boundaries (Section 4.3.2.2); and (iii) unidentified areas that are treated as the background. The first two groups of objects are treated in a similar way but with slight differences in the way that bounding rectangles are found and aligned. After the above matching objects have been identified, the reference frame background becomes the area that remains unidentified within the left frame. However, the right frame background is the area that remains non-coded after all the matched objects have been decoded at the encoder end. In order to prevent the transmission of shape information of the right frame objects and improve the compression...
efficiency, a special shape coding strategy is proposed (Section 4.4) in which the shapes of the right frame objects are determined by their matching left frame objects.

The coding strategy for a right frame object, which has been enclosed within its bounding rectangle, can be described as follows. The left object plane pixels that are outside the object area are first padded using a special Extrapolated Average Padding (EAPad) technique followed by a Extended Padding (EPad) technique, as described in Section 4.4.1. The interior blocks of the right frame objects are then encoded using a fixed disparity value, which is defined as the object disparity. This is found by calculating the average disparity of all internal blocks. If the mean squared error (MSE) of the internal blocks, calculated as the intensity difference between pixels, is lower than a certain threshold ($\text{MSE}=64$), we decide not to encode the errors. In such a situation, only the object disparity is transmitted to the decoder end. However, if the MSE is above the threshold, a further object extraction process is performed with new thresholds to identify more objects within the current object. The process is continued until the MSE for all identified objects fall below the threshold or the maximum number of iterations is achieved. In the latter case, the internal area errors are encoded using the pioneering block based prediction technique (see Section 3.3). This principle applies to the internal areas of all three groups of the objects.

The arbitrary shaped boundary blocks of the right object plane are encoded as described in Section 4.4.2. Briefly speaking, the best match for each boundary block is found from the left object plane, in which only the pixels that lie inside the right object are active in calculating the MSE. Once the location of the best match is found, the shape of the corresponding unpadded block in the left frame is used to determine the shape of the right frame boundary block. As a result of this process, the shape of the encoded right object is exactly the same as the matching left object. At the decoding end, however, the shape of the right object will be correctly reconstructed since the difference information will be encoded by other neighbouring objects or background, according to our object-matching scheme. To secure a high quality reconstruction at the decoding end, the disparity values and the predictive errors are always transmitted for all the boundary blocks. This also exploits the feature that human visual perception is more sensitive to boundary shape of objects than their internal areas where smoother texture is likely embedded. Finally, the
external blocks are not encoded as they fall into the interior of some other object or the background and would be coded accordingly.

In comparison with block based compression techniques, one of the advantages this scheme has is that, end-of-block codes (4 bits for each block usually) need not to be transmitted for every block in the right frame. This is due to the fact that most blocks will fall into the interiors of areas that need not be corrected. More details will be discussed in latter sections.

4.3 Contour Analysis of Stereo Images

This section proposes a contour analysis strategy for object extraction in stereo image pairs. It results in the development of a framework that enables matching objects to be identified in such image pairs. These objects are later used to exploit the redundancy present in stereo image pairs to achieve data compression.

4.3.1 Contour Extraction

Contour extraction is performed by a two-step process [161,165] popularly known as Caney edge detection. Firstly, the image is convolved with a Laplacian-of-Gaussian (LoG) operator [160,161]. The LoG of a continuous function \( f(x,y) \), \( [\text{LoG} \otimes f] \) is defined as follows:

\[
\text{LoG}(x,y) = \left( \frac{r^2 - \sigma^2}{\sigma^4} \right) \exp \left( - \frac{r^2}{2\sigma^2} \right)
\]

(4.1)

where, \( r^2 = x^2 + y^2 \) and \( \sigma \) is the standard deviation. To facilitate its application, equation (4.1) can be discretized in various ways [160]. As an example, Table-4.1 provides a \( 5 \times 5 \) LoG-mask with \( \sigma = 0.50 \).
After the image is convolved with a suitable LoG mask (note that the smoothness of the contours extracted will depend on $\sigma$), the edges are detected at the zero-crossing points (e.g., patterns such as "++--" and "--++" along both vertical and horizontal directions). In the second step, the slopes of the LoG of the image along both $x$ and $y$ directions, denoted by $S_x$ and $S_y$, are used to compute the edge strength at each zero-crossing point. An edge strength at a point is defined as follows:

$$s(x,y) = \begin{cases} +s & \text{if } (x,y) \text{ is a zero crossing point,} \\ 0 & \text{otherwise.} \end{cases}$$

The contour points are chosen based on the following criteria:

- The edge strength at each point along the contour is greater than $T_l$
- At least one point on the contour has an edge strength greater than $T_u$

where, $T_l$ and $T_u$ are pre-set thresholds and $T_l < T_u$.

Generally, $T_l$ is set sufficiently low to preserve the whole contour around the region boundary and $T_u$ is chosen large enough to avoid spurious edges. A contour search is initiated whenever one point with a value greater than $T_u$ is scanned. The search is conducted in both directions of the contour and the neighbouring pixels ($8$-connected) with values greater than $T_l$ are accepted as contour points. The search is terminated when no neighbouring pixels are found to satisfy this condition. Then all edge strength values along the detected contour are set to zero so that these points will not be visited again.
Chapter 4
An Object Based Algorithm for Stereo Image Pair Compression

The same search operation continues until the whole edge strength array has been scanned.

The parameters $\sigma$, $T_l$ and $T_u$ are chosen on an image pair basis. A candidate set of $\sigma(=0.5, 1, 2, 2.5)$ values were used to find the value for $\sigma$ that gives the best edge detection, for each stereo pair. The hysteresis thresholds are initially set at 30% ($T_l$) and 70% ($T_u$) between the maximum edge strength and minimum edge strength in the image. They are later adjusted so that the contour extraction process would result in an excess of 10 significantly sized (more than 128 pixels in length) contours for both images of the stereo pair.

The result of this stage would be the identification of a number of object contours from the image frames. Figure 4.2, shows an example of the identified objects in the image pair 'cans'. These contour plots are a result of using the function 'contour' in MATLAB [147] image-processing toolbox, which uses a strategy similar to the above in identifying object contours. Note that the short contours that cannot be used reliably in the matching process have been discarded.

![Figure 4.2 Left and Right Contour Plots of Image Pair 'Cans']
4.3.2 Contour Matching

After the above contour extraction process, the resulting contours are categorised into two types: those which are closed (Figure 4.3) and those which are open, which terminate at the image boundaries (Figure 4.4). As the matching procedures for the above two types of contours are different, they are treated separately in the following sections.

For the purpose of mathematical analysis we define a co-ordinate system with the origin at the bottom left hand corner of each image, with $x$-axis in the horizontal direction and $y$-axis in the vertical direction.
4.3.2.1 Closed contour matching

Closed contour matching is done in two steps. The first step is to calculate the following shape attributes for each contour.

(i) The number of pixels representing the perimeter of the contour, \( n \).
(ii) The \( y \) co-ordinate of the centroid \( (x_c, y_c) \).
(iii) The first invariant moment, \( h \).

Note that \( y_c \) is considered as a good matching attribute as we assume that the vertical stereo disparity between points in the image frames are negligible. This is due to the fact that parallel axis geometry has been used in obtaining all the test stereo image pairs.

If \( x_i \) and \( y_i \) represent the \( x \) and \( y \) co-ordinates of the points along the contour, the first invariant moment, \( h \) is defined as follows:

\[
h = \frac{1}{n^2} \sum_{i=1}^{n} [ (x_i - x_c)^2 + (y_i - y_c)^2 ]
\]

(4.3)

where, \( x_c = \frac{1}{n} \sum_{i=1}^{n} x_i \) and \( y_c = \frac{1}{n} \sum_{i=1}^{n} y_i \).

(4.4)

Every closed contour of the right frame is matched with every closed contour of the left frame. A contour from the left frame is accepted as an initial candidate match for the right frame contour, if the differences between each of their shape attributes fall below some pre-set thresholds (e.g. 20% for \( h \), and 10% for \( y_c \) and \( n \)).

At the end of this procedure, closed contours in right frames may have multiple candidate matches from the set of closed contours in the left frame. However, experimental results showed that such events were rare. Nevertheless in such circumstances a further step is necessary to find the best match among them. This is performed as follows.

Firstly, the contours are converted into their chain codes \( \{ a_i \in \{0,1,2,...,7\} \} \).
Chain Codes: A chain code is a more succinct way of representing a contour than a simple collection of co-ordinate points along the contour [61,66]. (e.g., \([x_1, y_1], (x_2, y_2), \ldots\]). It may be defined either with respect to the pixels or the boundaries between pixels. A pixel in a rectangular grid has either 4 or 8 neighbours depending on the definition of connectivity, as represented in Figure 4.5, below.

![Figure 4.5 Chain Code Representation](image)

The chain code is defined in this case by tracking around the white pixels in the outer boundary (as opposed for example to tracking round the black pixels in the inner boundary). For the examples shown, the codes are 1100100 etc. and 21017 etc [Note: one unit corresponds to an angle of 45°]. Alternatively, it is possible to track between the black and white regions along the intersections of the pixels using a 4-connected code. Thus a chain code describes the boundary as a series of one-pixel vector transitions at different orientations; only the starting point is defined explicitly.

A contour can be uniquely defined in 2D space, by its starting point and the chain code. However, chain codes can be made position independent if the start point is ignored, which simplifies matching of orientation-fixed linear figures. In the experiments performed, the leftmost out of the topmost pixel/s of a given contour is chosen as the
starting point. However, it is shown that the final result would be independent of this starting point.

The standard chain code representation above has certain drawbacks. For example a line along \(-22.5^\circ\) direction is coded as \(707070\ldots\). To prevent such a wraparound, Hui Li, B.S.Manjunath and S.K.Mitra [161] suggested the conversion of a length \(n\) standard chain code \(\{a_1, a_2, \ldots, a_n\}\) into a modified code \(\{b_1, b_2, \ldots, b_n\}\) by a shifting operation defined recursively by:

\[
\begin{align*}
    r_1 &= l_1 \\
    r_i &= q_i
\end{align*}
\]

where, \(q_i\) is an integer such that \((q_i - l_i) \mod 8 = 0\) and \(|q_i - b_{i-1}|\) is minimized, \(i = 1, 2, \ldots, n\).

The line along \(-22.5^\circ\) direction is then coded as \(\{787878\ldots\}\). The shifted chain code is further smoothed by a gaussian filter \(\{0.1, 0.2, 0.4, 0.2, 0.1\}\). In the work described below the above shifting and smoothing technique has been adopted.

Let \(\{l_i\}\) and \(\{r_i\}\) be the chain code representations of two contours \(L\) and \(R\), corresponding to the left and right frames respectively, and let \(N_L\) and \(N_R\) be their lengths. A measure of correlation \(MC_{kl}\) between two \(n\)-point segments, one starting at index \(k\) of contour \(L\) and the other starting at index \(l\) of contour \(R\) is defined as follows.

\[
MC_{kl} = \frac{1}{n} \sum_{j=0}^{n-1} \cos \frac{\pi}{4} (l'_{k+j} - r'_{l+j})
\]

where, \(l'_{k+j} = l_{(k+j) \mod N_L} - \frac{1}{n} \sum_{j=0}^{n-1} l_{(k+j) \mod N_L}\), and \(r'_{l+j} = r_{(l+j) \mod N_R} - \frac{1}{n} \sum_{j=0}^{n-1} r_{(l+j) \mod N_R}\), \(0 \leq i < n\).

Here, the modulus operation accommodates the cases in which the contours are closed. The correlation measure is similar to the mean-squared-error of two signals. The cosine function ensures that \(MC_{kl} \leq 1\), and \(MC_{kl} = 1\) when there is a perfect match. However, due to noise effects, our selection strategy of the starting point of the contours, occlusion
effects, inaccurate camera geometry (and the resulting relative rotation between the two contours), it is highly unlikely that there would be a one-to-one correspondence between the points having the same index, even in a matching contour pair. In addition the matching contours may be of a different length. The geometrical nature of the function $MC_{kl}$ and the search procedure for the orientation of the two contours described above deals with these issues.

In order to identify the location of the best fit between the two contours, an $n$-point segment of $R$, starting at index $k$, is slid over contour $L$. The similarity function, $F_{LR} = \max \{MC_{kl}\}_{k \in M}$, where $M$ specifies the search range, is then used to locate the best fit.

For a pair of closed contours $L$ and $R$, the whole contours can be used for the matching purpose. Suppose that $k = 0$, $n = N_L$, and that $M$ includes every index of contour $R$. Then the similarity function between contours $L$ and $R$ becomes:

$$F_{LR} = \max \{MC_{kl}\}_{0 \leq k < L}$$

Note that any rotation between the contours is reflected in the difference between the average values of the corresponding chain code representations. Since $MC_{kl}$ is normalised with respect to the mean value, the similarity function $F_{LR}$ is invariant to rotation if the quantization effect of the chain code is neglected. In implementation, the chain code of the longer closed contour was simply resampled by linear interpolation to have the same number of points as the shorter closed contour. Subsequently, the similarity criterion $F_{LR}$ was computed based on the resampled chain codes of the same length.

Contour $R$ in the right image and contour $L$ in the left image are selected as a matched pair, if the following two conditions are satisfied.

1. $F_{LR} \geq F_{L' R'}$, where $L'$ represents all the contours with similar shapes to contour $R$.
2. $F_{LR} > T$, where $T$ is a pre-set threshold that eliminates matches with poor correlation.
In the case that multiple contours get matched to the same contour, the pair with the highest $F_{lr}$ is selected.

### 4.3.2.2 Open contour matching

For matching of open contours, the salient segments along the contours are used as the matching primitives' [161]. Salient segments such as corners can be detected from the chain code representation. For a contour of length $n$ with chain code $\{l_i\}$, we define a measure of curvature at the $i^{th}$ point as,

$$C_i = \max_{1 \leq j \leq 3\sigma} \left\{ \max \left\{ \frac{1}{\sigma} \left| l_{i-j} - l_{i+j} \right|, \frac{1}{\sigma} \left| l_{i-j} - l_{i+j-1} \right| \right\} \right\}$$

The $i^{th}$ point along the contour is chosen as a salient point if both of the following conditions are satisfied,

1. $c_i \geq T_s$, and
2. $c_i \geq c_k$ for all $k \in [i-p, i+p]$.

where $p$ is a constant that determines the minimum distance between the salient points, and $T_s$ is a threshold specifying the minimum acceptable curvature. For example if $T_s = 2$, then the salient feature locations correspond to corners where the curve bends by at least $90^\circ$. The contour segments surrounding the salient points are then used as 1-D templates in finding the corresponding matches in the other image.
4.3.3 Contour Blocking

After a matching pair has been found, the next step is to enclose each contour within its smallest possible bounding rectangle. These rectangles are then extended in all four directions to lengths of integer multiples of 8 as such that they are centred at a modified centroid of the corresponding contour area. This enables the bounding rectangles to be sub divided into non-overlapping sub-blocks of size 8×8, for further processing. To facilitate the matching process, the size of the rectangle bounding the right contour is changed appropriately so that it is identical to the one bounding the left contour.

Let \( C = \{(x_i, y_i)\} \), where \( i = 1, 2, \ldots, n \), represents a \( n \)-point contour. The centroid of this contour \((x_c, y_c)\) is given by equation 4.4. We modify the y co-ordinate of this centroid to \( y_{cm} = \frac{1}{2}(y_{cl} + y_{cr}) \), where \( y_{cl} \) and \( y_{cr} \) are the y co-ordinates of the left and right contour centroids respectively. As we assume that parallel axis camera geometry and zero vertical disparity are used in obtaining the stereo image pairs, it is justifiable to align the bounding rectangles horizontally. Note that for the left contour we have: \( x_c = x_{cl} \), and for the right contour, \( x_c = x_{cr} \), where \( x_{cl} \) and \( x_{cr} \) represent the x co-ordinates of the left and right contour centroids.

In order to find the limits of the right object bounding rectangle, let us denote the top, bottom, left and right sides of the bounding rectangle by \( R_{top} \), \( R_{bottom} \), \( R_{left} \) and \( R_{right} \). Thus we have the following relationships:

\[
R_{top} = \max\{y_i\} + [8 - (\max\{y_i\} - y_{cm}) \mod 8] \tag{4.7}
\]

\[
R_{bottom} = \min\{y_i\} - [8 - (y_{cm} - \min\{y_i\}) \mod 8] \tag{4.8}
\]

\[
R_{right} = \max\{x_i\} + [8 - (\max\{x_i\} - x_c) \mod 8] \tag{4.9}
\]

\[
R_{left} = \min\{x_i\} - [8 - (x_c - \min\{x_i\}) \mod 8] \tag{4.10}
\]
Where, \( x_c = x_c, y_{cm} = \frac{1}{2}(y_c + y_{cr}) \) and 'mod' represents the mathematical operation \( \text{modulus} \).

Note that equations 4.7-4.10 would result in extending the tightest fitting rectangle to the object bounding rectangle, and in placing the modified centroid, \( (x_c, y_{cm}) \), on a vertex of a \( 8 \times 8 \) sub-block.

Figure 4.6, illustrates an example of the contour blocking process.

4.3.4 Block Classification

After enclosing the contour in the rectangle defined by equations 4.7-4.10, it is then divided into \( 8 \times 8 \) sub-blocks. The resulting sub-blocks can be classified into three types [Figure 4.7]:

- Blocks that lie fully inside the contour.
- Blocks fully outside the contour.
- Blocks on the boundary of the contour.
The classification of blocks is essential as they are treated differently in the coding process. The following strategy is used to differentiate the sub-blocks into the corresponding types.

Let \((x_{TL}, y_{TL}), (x_{TR}, y_{TR}), (x_{BL}, y_{BL})\) and \((x_{BR}, y_{BR})\) respectively be the co-ordinates of top-left, top-right, bottom-left and bottom-right vertices of a sub-block B. Note that \(y_{TL} = y_{TR} (= y_u)\), \(y_{BL} = y_{BR} (= y_l)\), \(x_{TL} = x_{BL} (= x_l)\) and \(x_{TR} = x_{BR} (= x_u)\).

Let \((x_k, y_k)\) be a point on a given contour C. We select B as a block which contains at least one point of C, if for some \(k (=1, 2, \ldots, n)\), at least one of the following four conditions are satisfied.

I. \(y_k = y_u\) \(\Rightarrow\) \(x_L < x_k < x_U\)
II. \(y_k = y_L\) \(\Rightarrow\) \(x_L < x_k < x_U\)
III. \(x_k = x_U\) \(\Rightarrow\) \(y_L < y_k < y_U\)
IV. \(x_k = x_L\) \(\Rightarrow\) \(y_L < y_k < y_U\)
Once the above boundary blocks are marked the rest fall into the two remaining categories: those that are completely inside the contour and those that are completely outside. If we find all four vertices of a sub-block to be inside the contour, we can guarantee that the block itself (i.e. all points belonging to the sub-block) would lie completely inside the contour. Similarly, if we find all four vertices to lie outside the contour, we can guarantee that the sub-block lies outside the contour.

4.3.5 Contour Filling Algorithm

Before encoding all those classified blocks, it is necessary to identify those pixel locations, which belong to the inside of the two contours. This is achieved by following a pixel based, parity check contour filling algorithm [158,159] for which a pseudo-code can be presented as follows. The idea is to scan pixels within an object bounding rectangle in a raster scan order, and while moving from the leftmost pixel in each scan line towards the rightmost, to count the number of times a boundary pixel was met. If the count is odd the pixel is taken as ‘belonging to the interior of the contour that represents the object’; else it is categorised as exterior.

1. For each $y$ do steps 2-11.

Begin
2. Set $count$ to zero
3. Set $x$ to the leftmost value of the grid.
4. While $x$ is less than or equal to the rightmost value of the grid do steps 5-11.
   Begin
5. If $(x,y)$ does not belong to the counter, then do steps 6 and 7.
   Begin
6. If $count$ is odd, then mark $(x,y)$ as belonging to the interior.
7. Increment $x$.
   End.
   Begin.
9. Call procedure LINK
10. If both above and below equal 1, then increment $count$. 
End.
11. If the sum above + below is not 0 or 2, then set the error flag.
   End.
   End.

End.

12. End of main algorithm

The pseudo-code for procedure LINK is given below.

Procedure LINK

0. Input \(\{x,y\}\in C\). Outputs, above and below.
1. \(\text{above}=0, \text{below}=0\)

2. If \(\{x-I, y+I\}\in C\), then \(\text{above}++\).
3. If \(\{x-I, y-I\}\in C\), then \(\text{below}++\).
4. While \(\{x,y\}\in C\) do steps 5-7.
   Begin.
5. If \(\{x,y+I\}\in C \& \{x-I, y+I\}\not\in C\), then \(\text{above}++\).
6. If \(\{x, y-I\}\in C \& \{x-I, y-I\}\not\in C\), then \(\text{below}++\).
7. \(x++\).
   End.
8. If \(\{x-I, y+I\}\not\in C \& \{x, y+I\}\in C\), then \(\text{above}++\).
9. If \(\{x-I, y-I\}\not\in C \& \{x, y-I\}\in C\), then \(\text{below}++\).
10. Return location of pixel \(\{x,y\}\) and values of counters above and below.

11. End of procedure.

The above algorithm proved to work well [158,159] for all those contours, which enclose full regions, i.e. the contour has no multiple points (no boundary doubles upon itself). However, there may be cases when the extracted contour does not enclose a full region. In such situations we either break the enclosed region into its constituent full regions and define each of them as a separate contour or disregard the non-full regions if they are insignificant. This is justifiable in extracting most objects.
An alternative approach: As the above parity-based, contour-filling algorithm fails when the enclosed region by the contour is not full, the ‘patch’ function provided with the MATLAB image processing toolbox [147] was used as an alternative for more accurate object boundary detection. The patch command fills a given contour (defined by \( \{x,y\} \) co-ordinate pairs of all the contour points) with a specified colour. An extension to this function was developed for the accurate detection of internal pixels, by reading the binary image produced by the ‘patch’ command.

Thus, if all four vertices \((x_{TL},y_{TL}),(x_{TR},y_{TR}),(x_{RL},y_{RL})\) and \((x_{BR},y_{BR})\) are inside the contour, we deduce that the sub-block defined by them lies completely inside the contour. Otherwise, they would be taken as being completely outside the contour. Note that, at this stage, there cannot be sub-blocks in which some of the vertices lie inside and the rest lie outside the contour. This is because we have already separated these sub-blocks using the algorithm proposed in Section 4.3.4.

4.4 Object Encoding

The stereo image pair is encoded on an object basis by identifying three types of objects within the images. They are:

- Objects that are bounded within closed contours,
- Objects that have open contours and both ends of the open contour terminate at image boundaries,
- The rest, which falls into the background.

Although there exist differences among the three types and their encoding requires individual co-ordination, the major proposed encoding techniques can be described as follows:

(i) Padding of left object bounding rectangles (hereafter referred to as OBR);
(ii) Disparity-based prediction for both arbitrary shaped boundary blocks and internal blocks;
(iii) Background encoding.
4.4.1 Padding of Left OBR

To match the arbitrary shaped objects inside the right frame, the left OBR need to be padded to ensure that all pixels inside the right object are covered and a full square error can be calculated to produce the best match. Specifically, each arbitrary shaped boundary block of the left OBR is padded using an extrapolated average padding (EAPad) technique. Firstly, the arithmetic mean value $A$ of all the block pixels $p(i,j)$ situated within the object region $L$ is calculated using the following formula:

$$A = \frac{1}{N} \sum_{(i,j) \in L} p(i,j)$$  \hspace{2cm} (4.11)

Where, $(1 \leq i, j \leq 8)$, $N$ is the number of pixels situated within the object region $L$. Note that the division by $N$ is done by rounding to the nearest integer. The next step is to assign $A$ to each block pixel situated outside of the object region $L$, i.e.

$$p(i,j) = A \quad \text{for all} \quad (i, j) \notin L$$  \hspace{2cm} (4.12)

After all the boundary blocks are padded according to the extrapolated average padding technique, the exterior blocks immediately next to the boundary blocks are filled by replicating upwards, downwards, leftwards and rightwards the row of samples from the horizontal or vertical border of the boundary macro-block having the largest priority number. The priority order is selected as shown in Figure 4.8.

---

**Figure 4.8 Priority of Boundary Macro-Blocks Surrounding an Exterior Block**

103
The remaining exterior blocks are filled with an intensity value of 128. At the end of this extended padding procedure \[146\] all the pixels within the left OBR are filled and it is ready to be used as the reference plane.

### 4.4.2 Disparity-Based Prediction

The arbitrary shaped boundary macro-blocks and the interior macro-blocks are encoded in different ways as described below.

#### 4.4.2.1 Encoding arbitrary shaped boundary macro-blocks

For boundary blocks, the best match of the right OBR is found by searching within a horizontally spread window, \( w = [-7, +7] \), centred at the corresponding block location in the padded left OBR. Note that the search needs to be done only in a horizontal direction as we assume parallel axis camera geometry. This assumption also implies that the objects in the left frame only displaces to the right of the right frame objects. However, in the above search procedure it is necessary to do the search on either side of the corresponding location, due to the fact that the search is done on an already displaced left OBR. For the same reason, the search range can also be limited to low values. The window size is appropriately clipped when the search is close to the boundaries of left OBR, so that search is always carried out only inside the left OBR.

Let \( L \) and \( R \) represent the regions within the left (unpadded) and right objects, respectively. We define two binary object planes \( \text{Alpha}_L \) and \( \text{Alpha}_R \) as:

\[
\text{Alpha}_L(k,l) = \begin{cases} 
1 & \text{if } (k,l) \in L \\
0 & \text{otherwise}
\end{cases}
\]

and

\[
\text{Alpha}_R(r,s) = \begin{cases} 
1 & \text{if } (r,s) \in R \\
0 & \text{otherwise}
\end{cases}
\]

Where values of \((k,l)\) and \((r,s)\) are limited inside the left and right OBRs. Let \( R\_block = \{x_{ij}\} \) represent a right block to be encoded, and \( L\_block = \{y_{ij}\} \) be a block from...
the padded left OBR, a search inside the window \((1 \leq i, j \leq 8)\) would enable us to calculate the squared error (SE) between the two blocks as follows.

\[
SE(R_{\text{block}}, L_{\text{block}}) = \sum_{i,j=1}^{8} (x_{ij} - y_{ij})^2 \times \text{Alpha}_R
\]  

As explained earlier, \(\text{Alpha}_R\) is used to ensure that only the pixels within the right object contribute to the calculation of the squared error. Hence, the best match, \(L_{\text{block_{match}}} = \{m_{ij}\}\), can be found by:

\[
L_{\text{block_{match}}} = \min_{L_{\text{block}} \in W} SE(R_{\text{block}}, L_{\text{block}})
\]  

Since the right OBR is not padded, each boundary block taken from the right OBR actually contains all the pixels whether they are inside the right object or not. If we represent this block by \(S = \{s_{ij}\}\), the error block between the left and the right can then be produced as follows:

\[
e(i, j) = (s(i, j) \times \text{Alpha}_L) - (m(i, j) \times \text{Alpha}_L)
\]  

In this equation, \(\text{Alpha}_L\) is used to determine the shape of those right objects. Consequently, the shape of the right object encoded in this way would be the same as that of its matching left object. Otherwise, we would need extra bits to encode the shape information for the right objects. However, this does not mean that we would have the shape of all right objects be distorted. As a matter of fact, the difference between the shapes of left and right objects would not disappear. They would be encoded by other neighbouring objects or background.

All the operations explained above in object extraction and encoding are carried out in reconstructed left frames for both the encoder and decoder. Hence, correct decoding is guaranteed by using the JPEG decoded left frame as a reference to reconstruct the right frame. The proposed algorithm can be applied in conjunction with any other conventional still image compression technique instead of being limited to JPEG. In addition, the disparity values and the prediction errors for each arbitrary shaped boundary block is
transmitted on an object by object basis, thus enabling the error blocks and disparity values to be properly identified and used to reconstruct the right objects at the decoding end.

4.4.2.2 Encoding interior blocks

For all right OBR blocks that are classified as interior, an MSE-based block matching technique is firstly used to find the individual disparity values. Secondly, the average of these individual disparity values is calculated and denoted as the object disparity value (i.e., a single disparity for the entire internal area of each object). Consequently, with this single object disparity value, matching blocks are selected to produce error blocks for each interior sub-block of the right OBR. The internal texture is detected by comparing the average error thus formed, with a selected threshold. If the error falls below the threshold, a decision would be taken not to encode the internal error blocks. However, the object disparity value, is transmitted along with other object parameters, in order to decode the internal right blocks.

If the internal texture is not smooth, the decision would be either to go into further levels of object extraction or to use the pioneering block based technique to encode those internal blocks. In this way, only error blocks are encoded without sending any disparity values.

The blocks which do not fall into either of the above categories, i.e. those that are in the exterior of the right object but within the right OBR are not encoded. They would be categorised as either background or a part of another extracted object, which would be encoded accordingly.

4.2.2.3 Background Encoding

The backgrounds of the two frames are defined as the areas that remain uncoded after all the matching object pairs have been selected and coded. The coding procedure for the background area of the right frame is similar to the coding procedure of its foreground objects. However, the bounding rectangles of the background areas of the stereo image pair would simply be the image boundaries.
4.5 Experimental Results and Analysis

To assess the proposed algorithm, extensive experiments were carried out by using the Disparity Compensated Transform Domain Predictive (DCTDP) coding algorithm proposed by M.G.Perkings [17] as our benchmark. The software simulation was implemented in MATLAB [147]. In order to make the experiments verifiable by other researchers, we established the test data set from a group of seven stereo image pairs downloaded from the Internet [84,148,149]. All the experimental results for the compression of all those right frames inside the data set are illustrated in Table-4.2.

Since the right frame is compressed in addition to the conventional JPEG, we use a so-called Extra Compression Ratio (ECR) (see Section 2.6.1) to measure the performance of the two algorithms tested. For the left frames, the compression performance is entirely dependent on JPEG, which is used as the baseline technique in this occasion.

<table>
<thead>
<tr>
<th>Image Pair</th>
<th>DCTDP Algorithm</th>
<th>Proposed Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ECR %</td>
<td>PSNR-dB</td>
</tr>
<tr>
<td>Cans</td>
<td>38.88</td>
<td>39.52</td>
</tr>
<tr>
<td>Lamp</td>
<td>35.17</td>
<td>41.38</td>
</tr>
<tr>
<td>Lab</td>
<td>40.02</td>
<td>33.19</td>
</tr>
<tr>
<td>Slanted</td>
<td>45.30</td>
<td>34.68</td>
</tr>
<tr>
<td>Packs</td>
<td>43.82</td>
<td>35.11</td>
</tr>
<tr>
<td>Texture</td>
<td>44.44</td>
<td>34.72</td>
</tr>
<tr>
<td>RISC</td>
<td>20.85</td>
<td>39.94</td>
</tr>
</tbody>
</table>

Table 4.2 Experimental Results – Object Based Coder

The total number of bits which are necessary to reconstruct the right frame contains overhead bits in the form of disparity values of boundary blocks, object disparity values, location information for the centroid of those right frame objects (only the x co-ordinate value) and the threshold values. The results shown in Table-4.2 for the proposed algorithm are produced by limiting the further thresholding of matched object internal
areas to one additional step. In other words, if the sub-objects formed by one more step of thresholding for the internal areas do not satisfy the matching criteria, the pioneering block based prediction scheme (see Section 3.3) is used to encode the errors. A search window size of $[16$ (horizontal), 0 (vertical)] has been used for both algorithms. The PSNR values quoted are calculated with respect to the original images.

![Figure 4.9 Reconstructed Image Samples:](image)

(a) by the benchmark algorithm  (b) by the proposed algorithm

The experimental results in Table-4.2 clearly show a significant improvement on the compression performance for the proposed algorithm against the benchmark. For all the test images, the proposed algorithm overwhelmingly outperforms the benchmark in terms of ECR values. It can be seen that two major factors contribute to this improvement. The first one is the fact that the predictive errors for the internal areas of those identified objects are not encoded if the matching texture is detected to be smooth and within our quality threshold. The second factor is that, as the constituent internal blocks are coded as single unit, the so called 'end of block' codes (usually a 4 bit special code word) that are necessary when the DCTDP coding scheme is used, is not needed for such smoothly textured areas. Note that, for any exclusively block based algorithm, the variable length of DCT error blocks require a special code word to indicate the end of each block, even if the predictive errors are all zeros. Table-4.2 also illustrates that the PSNR values obtained
for reconstructed right frames using the proposed algorithm are competitive in comparison with those achieved by the benchmark, although theoretically the PSNR values are smaller. This is caused by the selective coding strategy for the internal areas of objects. Figure 4.9 above illustrates an example of the right frame for Cans, in which part (a) is reconstructed by the benchmark algorithm and part (b) by the proposed algorithm.

Since we are using different coding strategies to compress those internal areas and boundary areas, visual comparisons between these two samples reveal that our reconstructed image has no noticeable difference compared to that of the benchmark. This is due to the fact that the proposed algorithm preserves the quality of reconstruction around the object boundaries, and would compensate for some loss of image quality in smoother internal areas of objects, as compared to the DCTDP scheme [17]. This quality preservation near object boundaries has been experimentally proved to be a basic necessity for the visual comfort of the viewers [31]. Essentially, the division of a boundary block into two blocks, which have smoother texture variation, together with the left object padding technique results in a more accurate boundary block being reconstructed.

<table>
<thead>
<tr>
<th>Image Pair</th>
<th>Improvement in ECR (%)</th>
<th>Improvement in PSNR (% dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cans</td>
<td>13.30%</td>
<td>-2.4%</td>
</tr>
<tr>
<td>Lamp</td>
<td>59.30%</td>
<td>-4.3%</td>
</tr>
<tr>
<td>Lab</td>
<td>12.40%</td>
<td>-4.9%</td>
</tr>
<tr>
<td>Slanted</td>
<td>15.2%</td>
<td>-3.0%</td>
</tr>
<tr>
<td>Packs</td>
<td>4.1%</td>
<td>-7.3%</td>
</tr>
<tr>
<td>Texture</td>
<td>28.90%</td>
<td>-3.6%</td>
</tr>
<tr>
<td>Risc</td>
<td>70.3%</td>
<td>-5.1%</td>
</tr>
</tbody>
</table>

Table 4.3 Percentage Improvements – Object Based Coder

To further analyse the experimental results in Table-4.2, we also present the improvement achieved by the proposed algorithm against the benchmark in Table-4.3. It can be seen that, in terms of ECR, the maximum improvement is 59.3%, which is indeed very significant. The minimum improvement, however, is only 4.1%. This further proves that
compression performance is always dependent on the compressibility and the nature of redundancy contained inside the input image. In terms of PSNR, all improvement values are negative, which means that the proposed algorithm is not able to outperform the benchmark.

Further observations reveal that the PSNR percentage figures are more convergent, compared with those ECR percentage values, which varies from a maximum $-7.3\%$ to a minimum $-2.4\%$. This explains that the distortion incurred in the proposed algorithm mainly comes from the strategy that when the matching texture is detected smooth and below a threshold, all errors are ignored. In other words, depending on the quality requirement, the distortion can be consistently controlled by the threshold.

### 4.6 Conclusions

In this chapter an object-based algorithm for the compression of stereo image pairs has been proposed. The basic idea behind the development of this algorithm is the efficient exploitation of redundancy in smoothly textured areas that are present in both frames, but are relatively displaced from each other due to binocular parallax. The identification and separation of such areas into matching object pairs improve the flexibility and efficiency when the coding strategy is decided. A special shape coding technique has been used for the boundary blocks in order to keep the coding efficiency to a maximum, yet enabling the right frame object shapes to be decoded using the shape of the matching left frame object. Experimental results show that significant compression improvement has been achieved in comparison with the existing block based coding techniques, while the reconstructed image quality remains competitive in PSNR measurement and visual inspection.

### 4.7 Improvements, Applications and Further Research

Potential for further research exists in the process of object matching and extracting in connection with disparity identification. In the present algorithm, we proposed a two-level object extraction scheme determined by the error texture (or matching texture between the two internal areas of objects). In other words, each time the matching texture is not smooth enough, only one more level of object extraction is allowed inside the
internal area. On the other hand, disparity is identified via the detection of the matching texture, in which we assume that the internal area would have consistent disparity if the matching texture were smooth. This pre-defined mechanism may not be optimal in the sense that smaller objects inside a larger object may not have consistent disparity. As the number of levels increases, however, compression efficiency would be affected negatively. How to make the best possible balance on an adaptive basis remains an important issue for further investigation.

In the present implementation of the algorithm, the object extraction has been based entirely on the detection of luminance gradients across prospective object boundaries. As a result of this, the objects that are recognised and classified as meaningful coding units may not represent realistic objects. For example two overlapping objects that are of similar surface texture may be considered as a single object even though the two objects would be at different depths (thus having different object disparities) from the camera image plane. This would either result in the non-selection of the objects as matching object pairs or in the loss of coding gain due to the excessive changes in the boundary shape. Although these occlusion effects are an inherent problem in stereo image compression, it is recommended that further refinements are incorporated [6,48] to the object extraction algorithm to address this issue. However, if the above object based stereo image compression algorithm was to be extended for stereo image sequence compression, the availability of motion information would provide a further dimension (the temporal) that could be exploited to aid in the extraction of more realistic video objects.

The proposed object based stereo image coding algorithm has a special advantage of being flexible in the selection of different levels of compression for different objects and areas, unlike the block based schemes discussed previously. It was mentioned that for more comfortable stereoscopic perception, edge information near disparity boundaries has to be preserved. The proposed algorithm has been designed taking this condition into consideration. The division of a boundary block into two blocks, which have smoother texture variation, results in a better quality edge reconstruction. As the internal areas of the foreground objects and the background are coded separately from the boundary areas it would be possible to maintain different levels of image quality on different objects/areas. Previous research has shown that in most applications the human
perception and attention would be centred on a dominant object on a scene. Thus, with the selective coding strategy adopted in the proposed technique and further modifications to the algorithm, it would be possible to maintain a higher quality for the dominant object whilst compensating the resulting loss of coding gain by compressing the less important areas more.

As the algorithm has been developed on an object-based platform, which is similar to that of MPEG-4, it can be transferable to the compression of stereo-videos or stereo image sequences. Under this circumstance, motion information in left frame sequence would need to be similarly exploited to determine the shape of object in the right frame sequence as well as the matching between each object pair. Further research is encouraged in extending the proposed algorithm to a MPEG-4 based stereo video coding algorithm.

The computational complexity of the proposed algorithm is mainly concentrated on the object extraction process. Further research is encouraged in the direction of speeding up these stages. However, the proposed algorithm provides a flexible approach towards stereo image pair compression and results in substantial coding gains, maintaining the image quality at equivalent levels to that of the state of the art stereo image compression techniques.
Chapter 5

Arbitrary Shaped Video Object Coding in MPEG-4

MPEG-4 [146] is the first audio-visual representation standard that understands a scene as a composition of audio-visual objects with a certain spatial and temporal behaviour. This chapter introduces the reader to the original contributions made by the author for improving some aspects of this standard.

The chapter is divided into three sections. Section 5.1 gives the basic concepts of the standard. Section 5.2 describes a way of using the contour based object extraction process introduced in Chapter 4 (see Section 4.3), as an alternative approach to various non-standardised segmentation based object extraction methods used in MPEG-4. Section 5.3 introduces the reader to a novel padding technique that can be used in the improvement of the coding efficiency of arbitrary shaped objects associated with MPEG-4 standard.

5.1 Introduction to MPEG-4 Standard

One of the most exciting characteristics of the world is the variety of forms, colours, and motions of physical elements with which we can interact or sometimes, just contemplate. For many decades, this rich variety of content has been observed through a rectangular window that presented a set of pixels, together with the associated audio. It is now time to have a closer look at the content and may even touch it! Recognising this fact, MPEG-4 will be the first audio-visual representation standard that understands the scene as a composition of audio-visual objects with a certain spatial and temporal behaviour.

Following the successful launch of MPEG-1 [140] and MPEG-2 [138,145], MPEG-4 [146] is now under development to provide standardised core technologies for efficient storage, transmission and interactive manipulation of video data to meet the ever growing demand of digital media services in multimedia environments. To secure the state-of-the-art technology
development in this new standard, MPEG-4 activities have been organised in terms of developing algorithms and tools for providing solutions to a number of key functionalities. These are expected to be the important issues that need to be addressed under the specifications. Typical functionalities associated with MPEG-4 include, content based efficient data representation & compression, object scalability, spatial and temporal scalability, error resilience etc.

With content-based data representation, a basic unit of AVO (Audio-Visual Object) represents audio-visual content. For visual information, an image frame is represented by a composition of video objects with a number of intrinsic properties, which includes shape, motion and texture. Separately coding these individual AVOs in a scene is a powerful tool that can remove a number of limitations inherent to current systems and standards. First it enables interaction with meaningful objects within the same scene. It also enables the re-use of data once the possibility exists to separately store and access objects, rather than frames. It will give the user the ability to create ones own content, by combining several of these stored objects in the same or different places. MPEG-4 transmits the composition information, which is necessary to form the scenes together with AVOs. The scene description information is encoded independently from those AVO bit streams and contains information about how objects are grouped together and are positioned in space and time etc. This independent encoding enables bit stream editing, i.e. one can change the composition of AV objects without having to decode their bit streams and change their content. If the position of the object was part of the objects bit stream, this would become very difficult.

5.1.1 MPEG-4 Video Objects

In section 5.1 it was shown that the philosophy of the MPEG-4 standard is based on AVOs. These in turn can consist of audio and video objects. For instance, a person in a video scene may be a video object, while the speech spoken by that person and the background noises can be audio objects.

A video object can be divided into a number of object layers to allow for spatial and temporal scalabilities [146,179]. For example in a video scene comprising of a person and a
vehicle, the person and vehicle could be defined as two separate object layers. Under each object layer, there is an ordered sequence of snapshots in time, which are referred to as Video Object Planes (VOPs). For instance in the above example the person taken as a separate entity from the rest of the frame in a sequence of video frames forms a VOP. These VOPs are the basic unit where MPEG-4 video compression is applied. A VOP is essentially a rectangular area that completely contains a video object but with the minimum number of macro-blocks contained within it. To preserve the content based functionalities of MPEG-4 the motion prediction and compensation is done based on these VOPs.

5.1.2 MPEG-4 Video Object Planes

In general, the input images to be coded in each VOP layer are of arbitrary shape and the shape and location of the images vary over time with respect to a reference window. For coding shape, motion and texture information in arbitrarily shaped VOPs, the MPEG-4 Video Verification Model (VM) [146] introduces the concept of a "VOP image window" together with a "shape-adaptive" macro-block grid. All VOP layers to be coded for a given input video sequence are defined with reference to a reference window of constant size. An example of a VOP image window within a reference window and an example of a macro-block grid for a particular VOP image are depicted in Figure 5.1.1, below.

In Figure 5.1.1, a VOP window with a size of multiples of 16 pixels in each image direction surrounds the foreground VOP of arbitrary shape and specifies the location of the macro-blocks, each of size 16x16 pixels. This window is adjusted to collocate with the most top and most left border of the VOP. A shift parameter is coded to indicate the location of the VOP window with respect to the borders of a reference window (original image borders).

The shape information of a VOP is coded (as binary Alpha planes – pixels inside VOP are represented by 1’s and rest by 0’s) prior to coding motion vectors based on the VOP image window macro-block grid and is available to both encoder and decoder. In subsequent processing steps, only the motion and texture information for the macro-blocks belonging to the VOP image are coded. These include the interior (standard) macro-blocks as well as the boundary (contour) macro-blocks indicated in the figure above.
5.1.3 Motion Estimation and Compensation

The MPEG-4 VM employs block-based motion estimation and compensation techniques to efficiently explore temporal redundancies of the video content in the separate VOP layers. In general, the motion estimation and compensation techniques used can be seen as an extension of the standard MPEG block matching techniques towards image sequences of arbitrary shape.

To perform block based motion estimation and compensation between VOPs of varying location, size and shape, the shape-adaptive macro-block grid approach for each VOP image is employed. Those macro-blocks that are completely outside the video object but belong to
the VOP are not coded as they belong to a different object or the background. A block-matching procedure is used for interior macro-blocks. The prediction error is coded together with the macro-block motion vectors used for prediction. The definition of the motion estimation and compensation techniques is, however, modified at the borders of a VOP. An image padding technique (Section 5.1.5) is used for the reference VOP frame $N-I$, which is available to both encoder and decoder, to perform motion estimation and compensation. After padding the reference VOP in frame $N-I$, a "polygon" matching technique is employed for motion estimation and compensation. A polygon defines the part of the boundary macro-block that belongs to the active area inside of the VOP frame $N$ to be coded and excludes the pixels outside of this area. Thus, the pixels not belonging to the active area in the VOP to be coded are essentially excluded from the motion estimation process.

5.1.4 Texture Coding

The Intra VOPs as well as the residual errors after motion compensated prediction are coded using a DCT on 8x8 blocks similar to the standard MPEG and H.263 standards. Again, the adaptive VOP window macro-block grid is employed for this purpose. For each macro-block a maximum of four 8x8 Luminance blocks and two 8x8 Chrominance blocks are coded. Particular adaptation is required for the 8x8 blocks straddling the VOP borders. The image padding technique (Section 5.1.5) in the figure above is used to fill the macro-block content outside of a VOP prior to applying the DCT in Intra-VOPs. For the coding of motion compensated prediction error $P$ (predicted) or $B$ (bi-directional) VOPs the content of the pels outside of the active VOP area are set to 128. Alternatively a low complexity shape-adoptive DCT (SADCT) technique is used to only encode the pixels belonging to the VOP - this results in higher quality at same bit rate at a slightly increased implementation complexity. Scanning of the DCT coefficients followed by quantization and run-length coding of the coefficients is performed using techniques and VLC tables defined with the MPEG-1/2 and H.263 standards, including the provision for quantization matrices. An efficient prediction of the DC- and AC-coefficients of the DCT is performed for Intra coded VOPs.
5.1.5 Reference VOP Padding

The boundary macro-blocks of the reference VOP are padded by a three stage padding process, as illustrated in Figure 5.1.2. The first step is to pad the boundary macro-blocks using Horizontal Repetitive Padding [HRPad]. Each sample at the boundary of the VO is replicated horizontally to the left and/or to the right direction in order to fill the transparent region outside the VO of the boundary macro-block. If there are two boundary sample values for filling a sample outside of a VO, the two boundary samples are averaged.

![Figure 5.1.2 MPEG-4 Padding Process](image)

In Figure 5.1.2 the horizontal-padded macro-block, \textit{hor}_\text{pad}[y][x] is generated by any process equivalent to the following example. For every line with at least one shape sample \(s[y][x] = 1\) (inside the VOP):

\[
\text{for } (x = 0; x < N; x + + ) \{
\quad \text{if } (s[y][x] = 1 ) \{ \text{hor}_\text{pad}[y][x] = d[y][x]; s'[y][x] = 1; \}
\quad \text{else } \{
\quad \quad \text{if } (s[y][x'] = 1 \& \& s[y][x''] = 1 ) \{
\quad \quad \quad \text{hor}_\text{pad}[y][x] = (d[y][x'] + d [y][x''])/2;
\quad \quad \quad s'[y][x] =1;
\quad \quad \} \text{ else if } (s[y][x'] = 1 ) \{
\quad \quad \}
\]
where $x'$ is the location of the nearest valid sample ($s[y][x'] = 1$) at the VOP boundary to the left of the current location $x$, $x''$ is the location of the nearest boundary sample to the right, and $N$ is the number of samples of a line. $s'[y][x]$ is initialised to zero.

The remaining unfilled transparent horizontal samples are padded using vertical repetitive padding [$hv$ pad], which is essentially the horizontal padding technique, performed in a vertical direction. ‘$hv$ pad[y][x]’ is generated by any process equivalent to the following example. For every column of $hor$ pad[y][x]:

```plaintext
for (y = 0; y<M; y++) {
    if (s'[y][x] = = 1) hv_pad[y][x] = hor_pad[y][x];
    else {
        if (s'[y][x] = = 1 & & s''[y][x] = = 1)
            hv_pad[y][x] = (hor_pad[y'][x] + hor_pad[y''][x])/2;
        else if (s''[y][x] = = 1) hv_pad[y][x] = hor_pad[y'][x];
        else if (s''[y'][x] = = 1) hv_pad[y][x] = hor_pad[y''][x];
    }
}
```

where $y'$ is the location of the nearest valid sample ($s'[y'][x] = = 1$) above the current location $y$ at the boundary of $hv$ pad, ‘$y''$ is the location of the nearest boundary sample below $y$, and $M$ is the number of samples of a column.

After all the boundary macro-blocks have been padded, the exterior macro-blocks immediately next to them are filled by replicating the samples at the border of the boundary
macro-block (*Extended Padding* [EPad]) that has the highest priority number (see Figure 5.1.3). The remaining exterior macro-blocks which are not located next to any boundary macro-block are filled with the value, 128.

![Diagram of Priority of Boundary Macro-Blocks](attachment:image.png)

This padded VOP is then used as a reference for motion compensation and prediction of arbitrary shaped boundary macro-blocks of the VO to be encoded.
5.2 A Contour Analysis Based Object-Extraction Technique for MPEG-4

Video Objects may be extracted from a scene by segmentation processes that can be performed off-line (non-real-time) or on-line (real-time), by automatic tools or semi-automatically. Though some segmentation techniques have been under study within and outside MPEG, none of these have been made a part of the MPEG-4 standard.

5.2.1 A Survey of Segmentation Based Object Extraction Techniques

During the development stages of the MPEG-4 standard [146], which concluded in November 1995, several proposals for video object extraction were submitted and investigated in depth by the MPEG-4 working group [137]. Following paragraphs are summaries of the most promising techniques. However, note that object extraction techniques are not a part of the standard, itself.

Microsoft and SESAME proposed semiautomatic segmentation techniques that allow the user to interact with the segmentation process by defining a rough outline of the objects contours (or by inserting some markers), which are then automatically refined (or expanded) and tracked through the sequence.

Some proposed algorithms adapted their own representation structure to pre-defined segmentation masks provided with MPEG-4 sequences. Proposals forwarded by EPFL, Mitsubishi and Motorola adapted a quad-tree structure to the given masks. In order to achieve better coding efficiency, several regions are identified inside each of the pre-defined segments. These methods use a hierarchical representation to simplify the selection of the best coding solution under rate-distortion constraints. The proposals by NEC and Microsoft divide the images into triangular patches that adapt themselves to the boundaries present in the pre-defined segmentation masks. These techniques involve an analysis step for feature point extraction (to identify the optimal triangular mesh node positions). The grid points close to the existing boundaries are modified to fit them.
A method proposed by SESAME automatically segments the frames into arbitrary shaped regions in a two step approach. The first is region tracking that tries to follow the time evolution of the regions and the second step is, creation of a partition tree where new regions can be introduced and small regions eliminated. Region tracking is based on the extraction of a set of markers for the moving objects, followed by a watershed morphological segmentation of the image to precisely define the region's boundaries. The partition tree starts from the tracked regions and can grow to higher level representations by merging regions with similar motion, or to lower level representations by splitting regions of inhomogeneous texture or motion. The resulting partition tree is the basis for coding. The appropriate level of details for each region has to be defined by a coding decision block (based on a rate-distortion criterion).

Some proposals base segmentation only on motion homogeneity. A method proposed by the University of Hanover, consists of three modules namely: change detection (frame difference thresholding) plus simplification using morphological operators; adaptation to spatial gradients (to improve boundary accuracy) and removal of uncovered background. A proposal from Thompson Inc., obtains time coherence of the arbitrary shaped regions by using an affine motion model.

AT&T proposed a hierarchical algorithm that locates, in an edge image, both head outline (modelled as an ellipse) and rectangular “eye-nose-mouth” regions. The algorithm consists of two steps namely: head outline detection by maximisation of a fitness ratio; searching for areas of maximal symmetry across a vertical or slightly slanted axis corresponding to facial symmetry, typically found in human faces. The output is a block-based segmentation of each video frame allowing a better distribution of the bit rate among different image areas.

In this section, the feasibility of using the novel contour analysis based object extraction technique, discussed in Chapter 4, for MPEG-4 video object extraction is discussed.
5.2.2 Proposed Technique

Figure 5.2.1, gives a simplified block diagram of the proposed scheme.

![Figure 5.2.1 Block Diagram of the Object Extractor](image)

The process is identical to what was used in the extraction of objects in stereo image pairs in Chapter 4 and each stage of Figure 5.2.1 works in a similar manner. However, instead of centring the bounding rectangle (VOP window) of the extracted video object, at the centre of area (i.e. the centroid) of the selected object contour, it is placed so that it satisfies the MPEG-4 VM requirements (see Section 5.1.2). Equations 5.1-5.4 provide ways of calculating the limits of the bounding rectangle.

Let \( C = \{(x_i, y_i)\} \), where \( i = 1, 2, \ldots, n \), represent an \( n \)-point contour of an object. Let us denote the top, bottom, left and right sides of the enclosing rectangle by \( y = R_{\text{top}} \), \( y = R_{\text{bottom}} \), \( x = R_{\text{left}} \) and \( x = R_{\text{right}} \) respectively. Thus, we have the following relationships:

\[
R_{\text{top}} = \max\{y_i\} + [8 - (\max\{y_i\} - \min\{y_i\}) \mod 8] \quad (5.1)
\]

\[
R_{\text{bottom}} = \min\{y_i\} \quad (5.2)
\]

\[
R_{\text{right}} = \max\{x_i\} \quad (5.3)
\]

\[
R_{\text{left}} = \min\{x_i\} - [8 - (\max\{x_i\} - \min\{x_i\}) \mod 8] \quad (5.4)
\]
Note that the purpose of this study was to find out the feasibility of using the intensity based object extraction technique developed in Chapter 4 in object extraction related to MPEG-4 video. As a result no motion information of the objects have been taken into account in contour detection.

5.2.3 Experimental Results

To demonstrate the use of this technique, the video sequence ‘Claire’ has been used. The input is a sequence of five adjacent video frames from this video clip, starting from $t = 0$ to and ending at $t = 4$. Figure 5.2.2 illustrates the results of several stages of the object extraction process. Figure 5.2.2 (a) shows the image contour, the bounding rectangle (VO window) and the grid that divides the VOP into non-overlapping macro-blocks. A macro-block size of $8 \times 8$ has been used. Note the location of the VO window with respect to the image frame. Equations 5.1-5.4 have been used in the positioning of the VO window with respect to the topmost and leftmost points of the object contour in compliance with the MPEG-4 requirements.

Figure 5.2.2 (b) shows the selected VOP placed within the image frame (VO window boundary is not shown) with the constituent macro-blocks categorised into interior (darker shading) and exterior. This classification has been done using the technique described in Section 4.3.4.

Figure 5.2.2 (c) illustrates the shape of the selected object in binary form. This information is used in MPEG-4 for shape coding of VOs. Such shapes are defined as Alpha planes in MPEG-4 terminology.
Figure 5.2.2 (a) Detected contour & VOP grid  
(b) Classified blocks: \textit{interior} & \textit{boundary}  
(c) Shape of object
Figure 5.2.3 illustrates how an object can be divided several sub-objects to improve spatial scalability of the video object. The sub-division of the object on the left into that shown on the right and the remaining part (not illustrated) has been possible due to the luminance gradient present along the boundary of the two sub-objects. This sub-division allows the replacement of the *coat* and *hair* (if a further differentiation was possible between these two, due to the presence of a colour or texture variation) with different ones.

Figure 5.2.4 shows the result of extracting the same object (the figure of Claire) from the five consecutive image frames of the sequence ‘Claire’. The results of frame No. 030 are also illustrated. Note the high amount of redundancy present between two consecutive VOs in this video sequence.
Figure 5.2.4 Same VO Chosen from Different Frames
5.2.4 Conclusion & Further Research

In this section a contour analysis based object extraction process has been proposed as an alternative to segmentation based techniques. Experiments have been limited to the object extraction based on spatial information. The proposed technique is suitable for high-level object extraction but would not perform well in the extraction of low-level objects.

However, further enhancements are possible by taking the motion information into account in separating the video object layers from each other and the background.
5.3 Improvements to the VOP Padding Technique of MPEG-4

The main drawback of the VOP padding technique described in Section 5.1.5 lies in the fact that it does not make use of the trend of pixel value variations usually present near object boundaries, in padding the exterior pixels of the reference video object. Extensive experiments indicated that, within arbitrary shaped boundary macro-blocks, certain trends exist with respect to pixel value variation. Replicating the boundary values in padding these blocks (as done in MPEG-4) would break this trend at the VO boundary. This in turn would result in higher prediction errors for those pixel locations that are within the shape of the current block, but are outside that of the best match. Better approximations to the corresponding pixel values can be found by padding the exterior pixels of the reference VO based on this trend.

5.3.1 Introduction – The Geometry of Prediction

Figure 5.3.1 and Figure 5.3.2 illustrate the prediction geometry of a block to be encoded.

Figure 5.3.1 indicates the case in which the shape (shaded, active area) of the block to be encoded lies completely inside the shape of the reference VOP block found as the best match. Note that only the pixels inside the block to be encoded are considered as active in the
motion compensation process. Under the above geometrical condition, it is seen that the padding values of the reference VOP have no effect on the prediction error block. Thus, the coding efficiency of such blocks cannot be improved by improving the padding technique.

After the error block is formed as indicated in Figure 5.3.1 below, either SADCT (where only the active pixels inside a shape similar to that of the block to be encoded would be coded) or normal DCT (in which case the external pixels are padded with zeros, as illustrated in Figure 5.3.1) is used to code it.

Figure 5.3.2 illustrates the situation where the best matching block from the reference frame lies completely inside the block to be encoded. Note that under this geometrical condition, some of the active pixels that would be coded will depend on the padded pixel values (area marked with pixel values, \( Enc - Pad \)). Thus, a part of the overall improvements achieved in coding efficiency, by any modification of the padding technique are due to blocks having similar prediction geometry. In general a similar situation would arise for all pixel locations within the \( N \times N \) (\( N=8 \)) error block where the corresponding pixel in the block to be encoded is within it's shape where as the same pixel in the best matching block is an exterior pixel. Note that most matching pairs of blocks have a prediction geometry, which is a combination of the extreme geometrical conditions illustrated in Figure 5.3.1 and Figure 5.3.2. Such blocks also benefit from any improvement to the reference VOP padding values.
Chapter 5

Improvements to the VOP Padding Technique of MPEG-4

A mathematical explanation would be given as below.

Let $Enc\{n_{ij}\}$ and $Ref\{r_{ij}\}$ be the block to be encoded and the best match found for it from the reference frame (with only internal pixels active), respectively. Let $Pad\{p_{ij}\}$ represent the padding value block that provides the padding values for the exterior pixels of $Ref$. Let $Err\{e_{ij}\}$ be the prediction error block. Thus, the elements of the error block is given by,

$$ e_{ij} = \begin{cases} 
  n_{ij} - r_{ij} & \text{if} \quad (n_{ij} \in Enc) \& (r_{ij} \in Ref) \\
  n_{ij} - p_{ij} & \text{if} \quad (n_{ij} \in Enc) \& (r_{ij} \notin Ref) \\
  \text{inactive / zero} & \text{if} \quad (n_{ij} \notin Enc) 
\end{cases} 
$$

(5.3.1)

5.3.2 Some Important Observations

Several simulations were designed to observe the geometry of matching arbitrary shaped blocks, in encoding the boundary macro-blocks of the VOP to be encoded. Experiments were done on various video sequences, having varying amounts of motion, texture and video objects. The standardised MPEG-4 VOP padding technique was used for the padding of exterior pixels of the reference VO.

The following observations were made:

(a) Due to the high amount of redundancy present between two VOs of adjacent video frames, the shape mismatch between most of the arbitrary shaped matching boundary macro-block pairs is minimal. I.e., the non-overlapped area is minimal.

(b) When the condition in (a) is true, better coding gains can be obtained if more realistic padding values are applied to pixels ‘just’ outside the VO boundary, in the reference VOP. (see Section 5.3.3)

(c) In instances where condition (a) does not hold true, the standardised padding technique for reference VOs (see Section 5.1.5) performs poorly. The reason being that the replicated boundary values of the reference VO are not good approximations to the corresponding pixels in the block to be encoded, especially at locations further inwards from the boundary of the current VOP. (see Section 5.3.5)
The above observations have been used in Sections 5.3.3 & 5.3.5, in suggesting improvements to the existing padding technique.

5.3.3 Proposed Solution – Linear Extrapolated Padding

In this section the reader is introduced to a novel padding technique [A3,A8] in which a single iteration of a row based Linear Extrapolation Padding [LEPad] step, is used to predict those exterior pixels adjacent to the boundary pixels of the reference VO.

The algorithm is described in mathematical detail as follows:

In every row of a $N \times N$ boundary macro-block, all exterior pixels that are immediate neighbours of at least one boundary pixel in that row are detected and denoted as *projected pixels* [see Figure 5.3.3].

![Figure 5.3.3](image)

Figure 5.3.3  ● Interior Pixels  ◇ Projected Pixels

These projected pixel values would then be predicted by the proposed linear extrapolation scheme as follows:
Assume that ‘\(n\)’ (\(1 < n < N\)) consecutive pixels in a row, immediately either to the left or to the right of a projected pixel (e.g. pixel denoted A in figure 5.3.3), are within the video object. Let these ‘\(n\)’ pixel values be represented by \(P_n\). Let \(X_n\) represent the column numbers of these pixels relative to the column number of the projected pixel. We use a linear equation of the form \(P = AX + B\) to fit all the internal pixel values denoted by \(P_n\), in least squared error terms, as follows:

Prediction error for a single pixel \(P_i\) is given by,

\[
E_i = P_i - (AX_i + B) \tag{5.3.2}
\]

For total least squared error,

\[
\frac{\partial}{\partial B} \sum_{i=1}^{N} [P_i - (AX_i + B)]^2 = 0 \tag{5.3.3}
\]

and

\[
\frac{\partial}{\partial A} \sum_{i=1}^{N} [P_i - (AX_i + B)]^2 = 0 \tag{5.3.4}
\]

By simplifying the above equations (5.3.3 & 5.3.4), we arrive at the following matrix equation.

\[
\begin{bmatrix}
N \\
\sum_{i=1}^{N} X_i \\
\sum_{i=1}^{N} X_i \\
\sum_{i=1}^{N} X_i^2
\end{bmatrix}
\begin{bmatrix}
B \\
\sum_{i=1}^{N} P_i \\
\sum_{i=1}^{N} X_i P_i
\end{bmatrix} =
\begin{bmatrix}
\sum_{i=1}^{N} P_i \\
\sum_{i=1}^{N} X_i P_i
\end{bmatrix} \tag{5.3.5}
\]

After constants \(A\) and \(B\) are found from the above equation (5.3.5), the projected pixel value can be found using the equation \(P_i = AX_i + B\).
In cases where \( n = 1 \) (e.g. pixel denoted by D in Figure 5.3.3), the projected pixel value is taken as equal to the single interior pixel value. If a projected pixel is flanked by interior pixels both on the left and on the right (e.g. pixel C in Figure 5.3.3), the above process is performed in both directions and the average of the two resulting extrapolated pixel values are taken as the value of the projected pixel. When \( P_n \) are bounded by two projected pixels (e.g. pixels denoted by B in Figure 5.3.3) both projected pixels will be determined by the same linear equation.

After all the projected pixels are padded, the MPEG-4 horizontal and vertical padding steps are performed to pad the rest of the exterior pixels within the macro-block. Note that the projected pixels that were padded using linear extrapolation now act as the new boundary pixels. After all boundary macro-blocks are padded similarly, extended padding (see Section 5.1.5) is used to pad the remaining exterior macro-blocks within the VOP bounding rectangle.

Figure 5.3.4 shows the proposed linear predictor used in conjunction with the MPEG-4 padding procedure. Note that the only difference as compared to the block diagram in Figure 5.2 is the introduction of the LEPad stage before the HRPad stage.
5.3.4 Experimental Results & Analysis

Experiments were carried out to calculate the coding gains obtainable with the proposed technique. The test results taken at various temporal locations for five test sequences are shown in Table 5.3.1.

Table 5.3.1. Experimental Results – Linear Extrapolated Padding

<table>
<thead>
<tr>
<th>Sequence Name</th>
<th>Frame Number \ Sequence Name</th>
<th>Extra Compression Ratio, %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>000-001</td>
<td></td>
</tr>
<tr>
<td>Susie</td>
<td>3.6</td>
<td></td>
</tr>
<tr>
<td>Tennis</td>
<td>6.6</td>
<td></td>
</tr>
<tr>
<td>Car Phone</td>
<td>2.1</td>
<td></td>
</tr>
<tr>
<td>Salesmen</td>
<td>4.1</td>
<td></td>
</tr>
<tr>
<td>Miss. America</td>
<td>6.8</td>
<td></td>
</tr>
</tbody>
</table>

The compression ratio (CR), in Table 5.3.1 is defined as follows:

\[
CR = \frac{\text{Bou}_\text{bits_{mpeg4}} - \text{Bou}_\text{bits_{proposed}}}{\text{Bou}_\text{bits_{mpeg4}}} \times 100 \quad \% \quad (5.3.6)
\]

where \( \text{Bou}_\text{bits_{mpeg4}} \) and \( \text{Bou}_\text{bits_{proposed}} \) are the amount of bits necessary to code all the boundary macro-blocks using techniques in MPEG-4 verification model (VM) and the proposed linear extrapolation based method, respectively.
Chapter 5 Improvements to the VOP Padding Technique of MPEG-4

Results in Table 5.3.1 clearly indicate that the proposed modification enhances the performance of the MPEG-4 padding technique by significant proportions. Further experiments indicated that with linear extrapolation, the optimal results in terms of compression gain, are obtained when the amount of iterations are limited to one, i.e. only one set of extended pixels are padded using the novel technique and the rest of the pixels are padded using the standardised technique. This is justifiable as further padding steps, following a linear variation towards the exterior, may result in larger prediction errors for pixels further away from the boundary. The proposed technique was found to especially outperform the MPEG-4 technique on object boundaries where shape changes are minimal where as in others it performed equally well.

Figure 5.3.5  (a) VOP to be encoded, t = 001  
(b) Padded VOP of t = 000, using MPEG scheme  
(c) Padded VOP of t = 000, using proposed scheme
Figure 5.3.5 (a) illustrates a VOP to be encoded from the test video sequence 'Claire' and its reference VOP, padded using the MPEG-4 padding technique (Figure 5.3.5 (b)) and the proposed padding technique (Figure 5.3.5 (c)). A close investigation of the two padded VOPs indicate that the one padded using the proposed technique has better approximating padded pixel values, to the boundary pixel values of the VOP to be encoded. Note that the padded pixels using the proposed coder also indicate a better match to the pixel value variation up to and beyond the boundary of the VOP to be padded. The apparent darker (in general), padded pixel values obtainable with the proposed technique provides proof of this behaviour due to the fact that the dress worn by ‘Claire’ is black in colour. This was experimentally found out to be the main reason behind the compression gain obtainable using the proposed padding technique.

Experiments were performed to check the feasibility of using the proposed technique in the vertical direction as well, i.e. before MPEG-4, vertical padding is done. The results indicated that the extra coding gains obtainable are insignificant and not worthy of the added complexity.

5.3.5 Extrapolated Average Padding Technique

Further experiments performed using the MPEG-4 padding technique and the proposed linear extrapolated padding technique introduced in Section 5.3.3 indicated that both techniques did not perform well in encoding arbitrary shaped MPEG-4 video objects that have been severely distorted (or have changed shape severely) between consecutive video frames. Although such occurrences are rare in video sequences (see Section 5.3.2) a solution to the problem would be nevertheless useful.

Figure 5.3.6 illustrates the reason for the failure of the two methods discussed already, under the above-mentioned condition. Note, that in the non-overlapping area the pixel errors are the difference between the corresponding extrapolated boundary pixel values of the reference VOP (Pad) and the interior pixel values (closer to the boundary) of the boundary macroblock (Cur) to be encoded. If the non-overlapping area is large, the boundary pixel values of the reference VOP (or the linear extrapolated pixel values) may not be a good representation.
Chapter 5

Improvements to the VOP Padding Technique of MPEG-4

for the interior pixel values (especially for pixels further away from the boundary) of the arbitrarily shaped boundary macro-block to be encoded. Thus, the methods discussed above would fail to produce lower magnitude error blocks.

As a solution to this, an extrapolated average based padding technique is proposed for the reference VOPs. In the proposed padding technique, the horizontal and vertical repetitive padding steps discussed above are replaced by a novel extrapolated average padding technique as illustrated in Figure 5.3.7 below.

![Figure 5.3.6 Error Block Formation in Distorted Boundary Blocks](image)

![Figure 5.3.7 Extrapolated Average Padding Technique](image)
A mathematical approach to the padding technique is described as follows. Firstly, the arithmetic mean value $A$ of all the pixels $p(i,j)$ of the boundary macro-block situated in the interior of the reference VOP is calculated using the following formula:

$$A = \frac{1}{N} \sum_{(i,j) \in L} p(i,j) \tag{5.3.7}$$

where, $(1 \leq i, j \leq 8)$, $N$ is the number of pixels situated within the reference VOP. The division by $N$ is done by rounding to the nearest integer. The next step is to assign $A$ to each block pixel situated outside the object region $L$, i.e.

$$p(i,j) = A \quad \text{for all } (i,j) \notin L \tag{5.3.8}$$

After the boundary macro-blocks are padded according to the above technique, extended padding (see Section 5.1.5) is used to pad the exterior macro-blocks. Any exterior macro-block that does not get padded at the end of this stage would be padded using 128.

### 5.3.6 Experimental Results & Observations

Figure 5.3.8 (b) illustrates the results of padding the VOP (same as in Figure 5.3.4) using the extrapolated average technique mathematically defined above. It illustrates the fact that this padding technique would not perform well for such objects, where shape distortion between two consecutive frames is minimal.

Experiments were performed on carefully selected consecutive image frames that consist of objects that change shape by significant proportions. The results were analysed both block-wise and object-wise, and it was observed that the extrapolated average padding technique outperformed the other two techniques (the standard MPEG-4 technique and linear extrapolation technique proposed above) for such objects/blocks. However, it is not recommended as a stand-alone padding technique or a replacement to the existing MPEG-4 technique as it performs badly in normal video sequences where the objects shape changes are minimal between consecutive frames.
5.3.7 Conclusions & Further Research

A modification for improving the coding efficiency of boundary macro-blocks of MPEG-4 arbitrary shaped objects has been proposed. The proposed technique makes use of the trend of pixel value variation present in boundary macro-blocks to reduce prediction errors. Experimental results indicate that coding gains of up to 7% are obtainable with the proposed modification.
Although complex functions that approximate the pixel value variations more efficiently could optimise the performance of this modification we recommend the use of linear extrapolation as a less computationally intensive modification to enhance the performance of the presently standardised padding techniques.

The use of an extrapolated average padding technique for severely distorted video objects has also been investigated. The results indicated that this padding technique outperforms the MPEG-4 and the linear extrapolation based padding techniques, on video objects that show severe shape changes between consecutive frames.

The research carried out on the VOP padding technique has resulted in the identification of two novel padding schemes. The Linear Extrapolation based technique outperforms the MPEG-4 padding technique in general, while it performs exceptionally better in instances where the object shape change between two consecutive frames, remain minimal. The Extrapolated Average based padding technique on the other hand outperforms the MPEG-4 scheme, as well as the Linear Extrapolation based scheme, in coding severely distorted VOPs. Thus, an integrated scheme between the two proposed techniques would be a significantly better alternative to the currently standardised padding technique.
Chapter 6 - Conclusion

This thesis has reported on the design, development and analysis of three novel algorithms for stereo image pair compression, and two novel algorithms for the improvement of video object coding related to the MPEG-4 multimedia standard.

The first algorithm exploits the redundancy present in the disparity field of stereo image pairs by using a single bit to indicate whether or not two adjacent disparities are equal. If the disparities are different the new disparity value is coded independently from the first along with one extra bit to indicate that a change has occurred. If the two adjacent disparity values are equal, only one bit is used to indicate this fact. Experimental results have proved that an average 7% increment of the compression performance can be expected by using this novel coding technique in disparity compensation based predictive coding of stereo image pairs. Some important observations with regards to the disparity maps of stereo image pairs have been made in the design and test stages of this algorithm.

The second algorithm, which completely eliminates disparity field coding, has been designed to simultaneously exploit the intra-frame and inter-frame redundancy present in stereo image pairs. It has been developed in three stages. The first stage involves the selection of a single pioneering block (the block preceding the block to be encoded) to search for the best match from the reference frame. Subsequently, it selects the block to the right of this best match as the predictor for the block to be encoded. The second algorithm uses two pioneering blocks (the block in front and the block above the block to be encoded) in disparity compensation and prediction. Experimental results indicate that the two pioneering block predictive coding algorithm can achieve extra compression gains of up to 28% as compared to the benchmark used to compare the performance of the proposed algorithms. However, under high levels of JPEG image quality loss in the reference frame, the above algorithms cannot guarantee the reconstruction of the predicted frame. A modification to the basic encoder that uses a locally reconstructed left frame as the reference for prediction and an additional feedback loop that reconstructs the encoded right image block ‘on line’ at the encoder end, has been provided as a solution to this problem. Several researchers have further tested the two pioneering block algorithm and have recommended it as the best existing algorithm for very low bit rate stereo image
coding. The basic idea of pioneering block coding has opened up significant interest amongst researchers working in the filed of video coding, in attempting to eliminate motion vector field coding. At present, further research is being carried out in this respect, with encouraging initial results.

The third stereo image compression algorithm proposed in this thesis uses an object-based approach to achieve extra data compression. The basic idea behind the development of this algorithm is the efficient exploitation of redundancy in smoothly textured areas that are present in both images, but are relatively displaced from each other due to binocular parallax. The proposed object based stereo image compression algorithm has achieved extra coding gains of up to 20%. However, as a result of the selective strategy used in the coding of internal areas of foreground objects and the background, the PSNR values of the reconstructed images are less than that of the benchmark algorithm. Despite the above observation, it has been shown that the visual image quality is maintained as compared to that of the benchmark algorithm. This is due to the fact that the special coding strategy of the proposed algorithm preserves the quality of reconstruction around the object boundaries, and thus compensates for some loss of image quality in smoother internal areas. The algorithm needs improvements and further refinement with regard to meaningful object extraction. At present, the object extraction has been solely based on luminance intensity gradients.

All three of the proposed stereo image compression algorithms exploit the inter-frame redundancy present between the left and right image frames of a stereo image pair to achieve extra data compression. This thesis also reports the results of research work that has been carried out in inter-frame redundancy removal in the temporal dimension, with regards to video sequence coding.

An algorithm that improves the coding efficiency of boundary macro-blocks of MPEG-4 arbitrary shaped objects has been proposed. It makes use of the trend of pixel value variation often present in boundary macro-blocks to reduce prediction errors. Experimental results have indicated that coding gains of up to 7% is obtainable by the proposed modification.
Finally, a contour analysis based technique to extract video objects from video sequences, to be used in association with the MPEG-4 multimedia standard, has also been proposed as an alternative to the complex segmentation based techniques currently in use. However, the algorithm needs further development in order to be used efficiently in the temporal dimension.
References


References

References


References


148
References


References


References

[84] http://biron.usc.edu/~wwoo/Research/Stereo/


References

References


References


References


References


APPENDIX-I

Rate Distortion curve for the Pioneer block encoder
APPENDIX-II

Journal & conference papers resulting from the research work documented in this Thesis.

Refereed Journal Publications


Refereed Conference Publications


Accepted Conference Papers


Hard copies of the papers A1, A3, A6, A7 and A10 are provided with this thesis.